## Introduction

Word stress as a linguistic phenomenon is governed by *language-specific rules*. For example, English (like many other languages) prefers alternating binary rhythm: (‘....´) (avoiding two adjacent stressed syllables) and phonetic (avoiding two adjacent unstressed syllables).

- Can a finite-state automaton that describes these rules be induced from examples?

## Method

- Data from Heinz (2009)
  - 106 transducers accurately modeling 106 different stress grammars
- Transducers generate sequences (‘words’) consisting of ‘w’, ‘s’ (syllables), where
  - w= syllable weight (7=1–4), and
  - s= stress level (0=0–2)
- Generated strings are input to learner trying to induce original grammar
- Generalization over input occurs based on similar context

### Experimental manipulation

- How many syllables should left and right context be (k and l respectively)?
- Is the generated finite-state machine equivalent to the original transducer?

## Architecture

### Generator

1. Generate sequences breadth-first
   - Minimal number of sequences to cover all paths
   - Measured in either sequences (‘w’) or individual symbols (‘s’)
2. Generate at least minimal number of sequences plus variable surplus in range: 0, 10, 20, 30, 40, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500

### Learner

1. Build prefix tree of example sequences
2. Identify partitioning
   - Collapse nodes in prefix tree that share a certain context (left and right context encoded in the parameters k and l respectively)
   - Use Myhill-Nerode names (Nerode, 1957, 1958) to quickly identify merging context

## Results — logistic regression

### Dependent variable: learned?

<table>
<thead>
<tr>
<th></th>
<th>B (SE)</th>
<th>Lower Odds Ratio Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept***</td>
<td>3.47 (1.043)</td>
<td>1.448 1.000</td>
</tr>
<tr>
<td>Complexityw***</td>
<td>-0.488 (0.111)</td>
<td>0.614 0.826</td>
</tr>
<tr>
<td>K***</td>
<td>-0.189 (0.060)</td>
<td>0.826 0.838</td>
</tr>
<tr>
<td>Complexityw***</td>
<td>0.047 (0.005)</td>
<td>1.048 1.058</td>
</tr>
</tbody>
</table>

### WITH OPTIMAL SETTINGS: k = 2, l = 0

<table>
<thead>
<tr>
<th></th>
<th>B (SE)</th>
<th>Lower Odds Ratio Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept***</td>
<td>1.297 (0.184)</td>
<td>3.660 1.000</td>
</tr>
<tr>
<td>Complexityw***</td>
<td>-0.842 (0.061)</td>
<td>0.431 0.486</td>
</tr>
<tr>
<td>Complexityw***</td>
<td>0.215 (0.020)</td>
<td>1.240 1.290</td>
</tr>
<tr>
<td>Surplusw***</td>
<td>0.087 (0.001)</td>
<td>1.007 1.008</td>
</tr>
<tr>
<td>Surplusw***</td>
<td>0.000 (0.000)</td>
<td>1.000 1.000</td>
</tr>
</tbody>
</table>

### Effect of k and l on learnability

The effect of k and l on learnability is shown in the graph. The optimal settings are determined based on maximizing the likelihood of learning the stress patterns.

### Effect of available data on learnability

The graph shows the relationship between the number of available data points and the likelihood of learning the stress patterns. The model shows a positive correlation, indicating that more data leads to higher learnability.

## Conclusion

- Learning is moderately successful (48.30% of languages)
- Failures are explained well by complexity of a language’s stress pattern
- There is also a significant effect of the amount of data fed to the learner; but effect size is small

## References

