

Manual: The Gradient Lexicon and Phonology Learner

Claire Moore-Cantwell

October 24, 2022

1 Quick Start

2 Details of Theories Implemented

2.1 Preliminaries

2.1.1 Theories of Eval

2.1.2 Perceptron Learning

2.2 UseListed

UseListed is a theory that many researchers implicitly or explicitly use as a default approach to exceptionality in phonology, but as far as I know no learning model has been developed. The theory can easily be summarized as "We memorize exceptions."

This theory assumes two things:

1. We memorize morphologically complex words, at least sometimes
2. We can therefore produce morphologically complex forms in two ways:
 - **Composed** forms are created by accessing multiple morphemes and realizing them together according to the morphological and phonological grammar
 - **Listed** forms are accessed whole, and realized according to their lexical entry and the phonological grammar

A couple of examples:

In Tagalog, morphemes often undergo 'nasal substitution' in which the final nasal of a prefix coalesces with the initial consonant of the root, forming a single sound. In Tagalog, morphemes often undergo 'nasal substitution' in which the final nasal of a prefix coalesces with the initial consonant of the root, forming a single sound.

In Tagalog, morphemes often undergo 'nasal substitution' in which the final nasal of a prefix coalesces with the initial consonant of the root, forming a single sound.

In Tagalog, morphemes often undergo 'nasal substitution' in which the final nasal of a prefix coalesces with the initial consonant of the root, forming a single sound.

In Tagalog, morphemes often undergo 'nasal substitution' in which the final nasal of a prefix coalesces with the initial consonant of the root, forming a single sound.

- (1) a. **diníg** /paŋ+**diníg**/ → pan-**diníg** ASSIMILATION
audible *sense of hearing*
- b. **daláŋin** /i+paŋ+**daláŋin**+in/ → ?i-pa-**naláŋin**-in SUBSTITUTION
prayer *to pray*

If we have both a **Composed** form and a **Listed** form available, there are many different ways we could decide between them.

2.3 Lexically Indexed Constraints

2.4 UR-constraints

2.5 Representational Strength Theory

2.6 Gradient Symbolic Representations

3 Supplementary details

3.1 Dealing with hidden Structure

3.2 Simulating Gen

3.3 Applying constraints

4 Input file details

5 Classes and Methods