

Manual: The Gradient Lexicon and Phonology Learner

Claire Moore-Cantwell

October 26, 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.

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

These examples come from ?. In (1a), the underlying velar nasal assimilates in place to the following stop, but in (1b), the two sounds have completely merged, leaving an [n] with the nasality of the [ŋ] and the place of the [d].

As the example suggests, the phonological shape of the individual words cannot completely predict whether the nasal will substitute or just assimilate. There are lexical trends, but no categorical rules. Importantly, individual words do not vary, the variation is entirely from word to word.

A UseListed approach to this pattern would say that both **pan-dínig** and **?i-pa-nalájin-in** are memorized as their own lexical entries. When speakers want to say the meaning '*sense of hearing*', or '*to pray*', they access those meanings directly in their lexicon, and produce them according to the idiosyncratic phonological form that is listed - one with substitution, and one without.

A second example:

English comparatives come in two forms, the ‘periphrastic’, using *more*, and the ‘morphological’, using *-er*. Both are available for most adjectives: *fouler* and *more foul* are about equally attested in a corpus, for example. However, higher-frequency adjectives exhibit idiosyncratic preferences ?.

- (2) a. simpler (96%) \gg more simple (4%)
 b. more stable (98%) \gg stabler (2%)

While a variety of phonological factors condition the choice between these two versions of the comparative, *simple* and *stable* are similar on all relevant dimensions¹. English speakers therefore must memorize that the meaning *simple* + COMPARATIVE is pronounced with *-er*, while *stable* + COMPARATIVE is pronounced with *more*.

Unlike in the case of Tagalog nasal substitution, many individual words do vary, and low-frequency words seem to follow the predictions of a probabilistic grammar. Even extreme cases like those in (2) still exhibit the minority form occasionally. One way to imagine what is happening here is that speakers have memorized the majority form for both words, but every so often they fail to use that memorized form and compose the comparative on the fly instead.

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

Option 1: Always use the listed form, if available

Option 2: Do whatever is easier. If it is easy to find the listed form, use it, but if it is easier to compose, use the composed form.

Option 3: Directly compare the Composed and Listed derivations in the same tableau.

¹They are 2-syllable words with initial stress, ending in [l], and are about the same lexical frequency.

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