

Supplementary Materials 1.1: Description of the morpheme parser

Supplement to Yang Wang and Bruce Hayes, “Learning phonological underlying representations: the role of abstractness,” published in *Linguistic Inquiry*

1. Model description

The goal of the modeling described in this section is to input a paradigm of labeled inflected forms and output the correct parse of the words of the paradigm into their component morphemes — prior to any knowledge of the phonological system. The model assumes a set of input data like (1) in the main text, repeated below:

(1) *Pseudo-German input data*

a. [bet]	cat _{stem}	[beda]	cat _{stem} plur.
b. [mot]	dog _{stem}	[mota]	dog _{stem} plur.
c. [lop]	turtle _{stem}	[loba]	turtle _{stem} plur.
d. [pap]	soup _{stem}	[papa]	soup _{stem} plur.
e. [mik]	plane _{stem}	[miga]	plane _{stem} plur.
f. [bek]	beer _{stem}	[beka]	beer _{stem} plur.
g. [es]	wine _{stem}	[esa]	wine _{stem} plur.
h. [nur]	light _{stem}	[nura]	light _{stem} plur.
i. [to]	toe _{stem}	[toa]	toe _{stem} plur.

The goal is to affiliate every segment in the data to a morpheme. If we use integer indices to mark this affiliation (as in the output files for our code), the intuitively correct outcome would be as in (2):

(2) *Correct parse for Pseudo-German input data*

a. [b ₁ e ₁ t ₁]	cat _{stem1}	[b ₁ e ₁ d ₁ a ₂]	cat _{stem1} plur. ₂
b. [m ₃ o ₃ t ₃]	dog _{stem3}	[m ₃ o ₃ t ₃ a ₂]	dog _{stem3} plur. ₂
c. [l ₄ o ₄ p ₄]	turtle _{stem4}	[l ₄ o ₄ b ₄ a ₂]	turtle _{stem4} plur. ₂
d. [p ₅ a ₅ p ₅]	soup _{stem5}	[p ₅ a ₅ p ₅ a ₂]	soup _{stem5} plur. ₂
e. [m ₆ i ₆ k ₆]	plane _{stem6}	[m ₆ i ₆ g ₆ a ₂]	plane _{stem6} plur. ₂
f. [b ₇ e ₇ k ₇]	beer _{stem7}	[b ₇ e ₇ k ₇ a ₂]	beer _{stem7} plur. ₂
g. [e ₈ s ₈]	wine _{stem8}	[e ₈ s ₈ a ₂]	wine _{stem8} plur. ₂
h. [n ₉ u ₉ r ₉]	light _{stem9}	[n ₉ u ₉ r ₉ a ₂]	light _{stem9} plur. ₂
i. [t ₁₀ o ₁₀]	toe _{stem10}	[t ₁₀ o ₁₀ a ₂]	toe _{stem10} plur. ₂

The task at hand is to find, prior to phonological discovery, which morpheme each segment should be coindexed with. We offer one way to do this, which works for the data sets studied in this article. To our knowledge this is the first effort to achieve this goal, and it is likely that many alternatives might fruitfully be explored.

Our system is based on enumerating all the logically possible morpheme parses and choosing the best one as defined in a particular way.¹ The enumeration of candidate parses works as follows. If word w in the training data contains n morphemes, then for each segment in w there are n possible morphemic affiliations. If word w contains m segments and n morphemes, then the number of possible morphemic affiliations for w will be m^n . Then, for the full set of words w_i in the training data, we cross-classify all of these morphemic affiliations, obtaining $\prod_i m_i^{n_i}$ candidates.

As discussed in §5.1 of the main text, the best morpheme parse is likely to be the one that maximizes paradigm uniformity; i.e. minimizes the degree of divergence between the allomorphs of the morphemes present in the data. Thus, to evaluate any particular candidate, we assign it a penalty score that reflects this divergence. This score is built up piecewise from the segment level to the whole-parse level.

To start, we assess the dissimilarity of segment pairs. The difference is computed on the basis of their mismatching feature values, using a fairly standard phonological feature set. Each feature contributes a specific degree of dissimilarity penalty (for example, mismatches in [voice] contribute 3.4). The feature-specific penalties were established using the method of Wilson (2006) and White (2017): they employed the values that provide the best fit to confusion matrix data gathered experimentally (we used the data in Cutler et al. 2004). For any two segments, the dissimilarity is calculated as the sum of the penalties for every feature in which they differ.

The next task is to compute allomorph dissimilarity. We first compute for any given pair of allomorphs the optimal segment-to-segment alignment for that pair, using the method from Kruskal (1983) described in §5.3 of the main text. The method requires penalty weights for both (i) correspondence of non-identical segments, and (ii) correspondence of a segment to null. For (i), we used the values of segment dissimilarity just described, and for (ii) we adopted an *ad hoc* value, namely 45. A sample optimized alignment is shown in (3) below for the Pseudo-German allomorphs [bet] and [bed]. Here, the dissimilarity penalty comes solely from the difference of [t] and [d], 3.4. The dissimilarity of [apt] and [aptə], shown in (3b), derives from the segment-to-null penalty, 45.

¹ Our procedure is strongly reminiscent of classical (non-stochastic) Harmonic Grammar (Legendre et al. 1990), with the constraints drawn from the Correspondence Theory of McCarthy and Prince (1995). However, since our system is a procedure for morpheme discovery, not a phonological grammar, we present it here in neutral terms to avoid confusion.

(3) *Three representative best allomorph alignments with penalties*

<i>a. Pseudo-German ‘dog’</i>			<i>b. Epenthesis/Deletion</i>				
b	e	d	a	p	t	ə	
b	e	t	a	p	t	∅	
0	0	3.4	0	0	0	45	penalty weights
	3.4			45			total penalty

It should be clear that any other alignment for these cases would incur a much higher penalty; for instance a nonoptimal alignment for (3a), [b]-∅, [e]-[b], [d]-[e], [∅]-[t] (superfluous correspondence to null), would incur a very large penalty (124.9).

1.1 Defining similarity for complete candidate parses

Starting with the dissimilarity values for each possible pair of distinct allomorphs, we scale up as follows. The dissimilarity penalty for a *morpheme* is the average of the penalties for all the allomorph pairs in which it appears, scaled so as to penalize alternation in stems more severely. The dissimilarity penalty for a complete parse (i.e. of the whole dataset) is the sum of the penalties for the set of morphemes it contains (no frequency weighting). The similarity penalties for stems and affixes are employed as part of the process (see below) of finding the winning parse.

1.2 Variegation

We have found cases in which the system as just stated can be led astray. As an extreme case, imagine a language in which every noun must be preceded by a noun-class prefix, and the noun class prefixes alternate between [i] and [u] by a process of vowel harmony. Thus, for instance, the stem /kimen/ ‘cat’ appears as: singular [ni-kimen], dual [bi-kimen], plural [ri-kimen]; but /kurat/ ‘dog’ appears as [nu-kurat], [bu-kurat], [ru-kurat], due to vowel harmony. Here, there is a morpheme parse that is incorrect by the linguist’s criterion but has perfect similarity (each morpheme has one allomorph):

(4) *An incorrect parse with perfect allomorph similarity*

n-ikimen	‘cat sg.’	n-ukurat	‘dog sg.’
b-ikimen	‘cat dual’	b-ukurat	‘dog dual’
r-ikimen	‘cat pl.’	r-ukurat	‘dog-pl.’
n-ipetep	‘bird sg.’	n-uloran	‘horse sg.’
b-ipetep	‘bird dual’	b-uloran	‘horse dual’
r-ipetep	‘bird pl.’	r-uloran	‘horse pl.’

What we want instead is given in (5).

(5) *Correct parse*

ni-kimen	‘cat sg.’	nu-kurat	‘dog sg.’
bi-kimen	‘cat dual’	bu-kurat	‘dog dual’
ri-kimen	‘cat pl.’	ru-kurat	‘dog-pl.’
ni-petep	‘bird sg.’	nu-loran	‘horse sg.’
bi-petep	‘bird dual’	bu-loran	‘horse dual’
ri-petep	‘bird pl.’	ru-loran	‘horse pl.’

This constructed cases is an extreme one, but it reflects a more general problem: without suitable intervention, the system is liable to avoid stem-conditioned alternation in affixes by assigning the alternating segments to the stems.

We find that bad parses like (4) can be located, and then harshly penalized, because they give rise to an excessively restrictive pattern of stem types: under the bad parse, every stem must begin with just one of two phonemes, /i/ or /u/. This is an unlikely pattern for real languages, where the inventory of possible stems is usually made abundant by allowing for many phonological possibilities. In our implemented system, we included a penalty based on what we call *Variegation*; it is calculated by assessing the inventory of possible stem-beginnings and stem-endings and penalizing those in which these inventories consist of very few sounds. We find that this principle can sufficiently penalize the parsed candidate in (4) to the point that the correct parse in (5) will win, even though it has some alternation in the prefixes; and similarly for other cases.

1.3 *Computing the overall penalty for a parse*

The overall penalty consists of a weighted sum of three factors: (a) the dissimilarity penalty for stem allomorphs; (b) the dissimilarity penalty for affix allomorphs; (c) the *Variegation* penalty. We set the weights ad hoc, in a way that yield the correct parses for the languages examined.

1.4 *Searching for the best parse*

The number of parses can in practice be very large (for the Catalan example in the main text, it is about 2×10^{161}). This precludes the strategy of examining the all, so instead we employ a greedy hill-climbing search in which a set of candidates, starting with a random seed, gets edited in various ways, with improved candidates kept.

(6) *Details of search*

- a. For monomorphemic words, fix their unique possible parse in place as the first step.
- b. Assign up front the index of a morpheme to adjacent strings of non-alternating sounds that always occur in the presence of that morpheme.
- c. Edits for the hill-climbing search:
 - change morphemic affiliation of individual segments (try each case)
 - swap morphemic affiliation of pairs of segments (try each case)

- extend a hypothesized stem allomorph, present in one form, to the whole paradigm

We employ a beam search. The hill-climbing process begins with a beam of candidates derived from step (6a), with unaffiliated segments given a random initial affiliation. Lastly, although morphemes with non-contiguous segments are included in the candidate set, the search is more efficient if it is biased to prefer contiguous morphemes (through a fourth penalty term, with an ad hoc weight).

This system suffices to obtain the intuitively correct morpheme parse for all of the language simulations mentioned in this article.

2. Directions for improvement

We noted in the main text that our system attempts to solve a problem not previously addressed in the computational literature, and we judge that further attempts to solve it would be beneficial. Improvements might be possible on several lines.

(a) The **metric of similarity** for allomorphs might be more accurate if it considered the *contexts* in which segments occur, since this seems to be right for similar problems in phonology (Steriade 2001, Fleischhacker 2005).

(b) **Discontinuous morphemes** (from infixation, metathesis, or root-and-pattern morphology) are, in our experience, harder to find, and sometimes defeat our model, rather badly in the well-known case of Semitic consonantal roots (e.g. Arabic *ḵatab*). Discontinuous morphemes also would require beefing up the morphological system of our main model to cover the assembly of discontinuous morphemes into words. For proposals in this area see Wilson (2018) and Xu et al. (2020).

(c) **Phonological coalescence**. Segments arising from phonological coalescence go unaddressed here. In our notation, coalesced segments would bear two morphemic indices, as in Sanskrit /raja₁ i₂ndra/ → [raje:₁i₂ndra] ‘supreme sovereign’ (de Haas 1988:1), but such cases are not included in our candidate sets.

(d) **Search**. It seems likely that our search for the best parse could be carried out more efficiently.

(e) **Multiple parses**. The system could be made more flexible by having it forward to phonological learning a *set* of likely parses rather than just one. These could make the system tolerant of error at a modest cost in total search space.

References

- Cutler, Anne, Andrea Weber, Roel Smits, and Nicole Cooper. 2004. Patterns of English phoneme confusions by native and non-native listeners. *Journal of the Acoustical Society of America* 116:3668–78.

- Fleischhacker, Heidi. 2005. Similarity in phonology: Evidence from reduplication and loan adaptation. Ph.D. dissertation, UCLA, Los Angeles, CA.
- de Haas, Wim. 1988. A formal theory of vowel coalescence: A case study of Ancient Greek. Ph.D. dissertation, University of Nijmegen.
- Kruskal, Joseph. 1983. An overview of sequence comparison: Time warps, string edits, and macromolecules. *SIAM Review* 25:201-237.
- Legendre, Geraldine, Yoshiro Miyata and Paul Smolensky. 1990. Harmonic Grammar – A formal multi-level connectionist theory of linguistic well-formedness: An Application. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, 884–891. Mahwah, NJ: Lawrence Erlbaum Associates.
- Steriade, Donca. 2001. Directional asymmetries in place assimilation: A perceptual account. *The role of speech perception in phonology*, ed. by Elizabeth Hume and Keith Johnson, 219-250. San Diego: Academic Press.
- White, James. 2017. Accounting for the learnability of saltation in phonological theory: A maximum entropy model with a P-map bias. *Language* 93:1-36.
- Wilson, Colin. 2006. Learning phonology with substantive bias: an experimental and computational study of velar palatalization. *Cognitive Science* 30:945-982.
- Wilson, Colin. 2018. Modeling morphological affixation with interpretable recurrent networks: sequential rebinding controlled by hierarchical attention. *CogSci 2018*: 2693–2698.
- Xu, Hongzhi, Jordan Kodner, Mitchell Marcus, and Charles Yang. 2020. Modeling morphological typology for unsupervised learning of language morphology.