Extracting morphophonology from small corpora

Marina Ermolaeva
University of Chicago / Chicago, IL, USA
mermolaeva@uchicago.edu

Abstract

Probabilistic approaches have proven themselves well in learning phonological structure. In contrast, theoretical linguistics usually works with deterministic generalizations. The goal of this paper is to explore possible interactions between information-theoretic methods and deterministic linguistic knowledge and to examine some ways in which both can be used in tandem to extract phonological and morphophonological patterns from a small annotated dataset. Local and nonlocal processes in Mishar Tatar (Turkic/Kipchak) are examined as a case study.

1 Introduction

Morphophonology, or the interface between morphology and phonology, encompasses a wide range of phenomena. While this paper primarily focuses on learning phonological rules from a dataset, it is difficult to draw generalizations based only on surface strings, since the rules may be morphologically specific. The challenge goes beyond learning which phonotactic sequences are allowed, also incorporating surface realizations of morphemes and rules governing their distribution.

Large unannotated corpora are used by a large portion of the existing work on learning phonological patterns e.g. approaches to learning vowel harmony (Goldsmith and Riggle 2012; Szabó and Çöltekin 2013; Flinn 2014). However, in the case of rare languages, a large corpus may be unavailable. On the other hand, *small* hand-annotated examples or texts are a natural output of linguistic fieldwork and readily available even for under-resourced and under-studied languages.

Interlinear glossed text is a format traditionally utilized in linguistic papers for presenting language data. It annotates each morpheme with a label, or *gloss tag*. When the amount of data is insufficient, the role of such linguistic knowledge in making generalizations becomes more prominent; see (Wax 2014; Zamaraeva 2016) for approaches to extraction of morphological rules that take this path.

When it comes to morphophonology, agglutinative languages are of special interest. They tend to exhibit a variety of interacting processes which give rise to multiple surface realizations of most morphemes.[1] A small dataset is very likely to contain only a subset of possible allomorphs – an additional challenge for the learning algorithm. As a case study, this paper focuses on Mishar dialect of Tatar language (Turkic/Kipchak). The data sample used here is a hand-glossed collection of texts elicited from native speakers in the course of fieldwork (MSU linguistic expedition 1999–2012) (3090 word tokens; 1740 types).

2 (Morpho)phonological processes

One common type of alternations stems from *local* processes, where the context is immediately adjacent to the segment undergoing the change. The same surface segment may arise from different processes. For example, the ablative suffix (1) has different realizations after voiceless consonants, nasals, and elsewhere. The locative morpheme (2) demonstrates only a two-way distinction after voiceless consonants and elsewhere. The plural suffix (3) is also sensitive to a two-way distinction, drawing a line between nasal consonants and other segments.

(1) a. kibet-**tän**
shop-ABL

b. kɤz-**dan**
girl-ABL

c. urɤn-**nan**
place-ABL

[1] For example, the idea of correlation between agglutination and vowel harmony goes back to (Baudouin de Courtenay 1876, 322–323), and its history and development are documented in (Plank 1998).

Proceedings of the 15th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 167–175
Brussels, Belgium, October 31, 2018. ©2018 The Special Interest Group on Computational Morphology and Phonology
https://doi.org/10.18653/v1/P17

(2) a. jɤrt-**ta** c. ten-**dä**
yard-LOC night-LOC

b. kɤz-**da**
girl-LOC

(3) a. at-**lar** c. ujɤn-**nar**
horse-PL game-PL

b. kɤz-**lar**
girl-PL

While this data does hint at certain general phonotactic patterns (e.g. a voiceless stop is never followed by an affix beginning with a voiced obstruent), the contexts cannot be inferred exclusively from surface strings; each morpheme has to be considered separately. Moreover, even the same set of alternants may be found in multiple processes. Consider the following voicing alternation:

(4) a. matur-**lɤg**-ɤ
pretty-NOMIN-P3

b. jaxšɤ-**lɤk**-ka
good-NOMIN-DAT

(5) a. kal-**gan**
stay-PFCT

b. čɤk-**kan**
exit-PFCT

The difference between (4) and (5) lies in the *directionality* of the {g, k} alternation: the obstruent in the former is located at the left edge of the affix and assimilated to the preceding segment; in the latter it is sensitive to the voicing of the following segment.

Another prominent source of allomorphy is vowel harmony. This process is *nonlocal* in the sense that it only affects a subset of segments (in this case, the set of vowels); all other segments are transparent and do not interact with the rule in any way. Vowel harmony can be analyzed of in terms of *underspecification* (Archangeli 1988): vowels in affixes lack some feature specifications, and their surface realization is dependent on the closest fully specified vowel.

In Mishar Tatar, vowels are subject to fronting harmony controlled by the root; most affixes have front and back allomorphs.

	[−BK,	[−BK,	[+BK,	[+BK,
	−RND]	+RND]	−RND]	+RND]
[+HI, −LO]	i	ü	ɤj	u
[−HI, −LO]	e	(ö)	ɤ	(o)
[−HI, +LO]	ä		a	

Table 1: Mishar Tatar vowels

(6) a. bala-lar-ɤbɤz-ga
child-PL-P1PL-DAT

b. täräz-lär-ebez-gä
window-PL-P1PL-DAT

These phenomena are not completely free of exceptions and problematic cases. Two instances of non-canonical vowel harmony in suffixes attached to borrowed roots are presented in (7). Similarly, (8a) shows the expected voiced variant of PFCT arising after a vowel while (8b) demonstrates the exceptional unvoiced variant in an identical phonological context. Another issue is true allomorphy triggered by morphosyntactic features as opposed to phonological context; and determining which is the case is in itself a nontrivial task. For example, in (9) 2PL is realized differently depending on the TAM (tense/aspect/mood) marker on the verb.

(7) a. tarix-**ɤ** b. činovnig-**ɤ**
history-P3 official-P3

(8) a. i-**kän** b. di-**gän**
AUX-PFCT speak-PFCT

(9) a. bar-a-**sɤz** b. bar-dɤ-**gɤz**
go-ST.IPFV-2PL go-PST-2PL

3 Finding alternations

3.1 Alternations as sets

Consider the following rule encoding Mishar Tatar vowel harmony:

(10) $$\begin{bmatrix}+\text{SYL}\\ 0\,\text{BK}\end{bmatrix} \rightarrow \begin{bmatrix}\alpha\text{BK}\end{bmatrix} \;/\; \begin{bmatrix}+\text{SYL}\\ \alpha\text{BK}\end{bmatrix}(\begin{bmatrix}-\text{SYL}\end{bmatrix})^{*} \;\_\_$$

(11) $$\{\text{e}, \text{ɤ}\} \rightarrow \text{e} \;/\; (\text{e}\,|\,\text{i}\,|\,\text{ä}\,|\,\text{ö}\,|\,\text{ü})\,(\text{b}\,|\,\text{d}\,|\,\text{g}\,|\,\ldots)^{*} \;\_\_$$
$$\{\text{e}, \text{ɤ}\} \rightarrow \text{ɤ} \;/\; (\text{ɤ}\,|\,\text{ɤj}\,|\,\text{a}\,|\,\text{o}\,|\,\text{u})\,(\text{b}\,|\,\text{d}\,|\,\text{g}\,|\,\ldots)^{*} \;\_\_$$

This rule can be represented succinctly using feature bundle notation (10): any vowel not specified for [±BK] receives these values from the closest vowel to the left. However, underspecified vowels can be equivalently thought of as sets of fully specified segments, and the rule as the condition determining which member of the set appears on the surface, e.g. (11). An *alternation* can then be defined as the set of all surface outcomes of a process, each associated with a set of contexts that trigger it.

At this proof-of-concept stage we adopt the following *simplifying assumption*: alternations occur between segments (i.e. one-segment substrings), or between a segment and zero, and the context of each alternant is a segment, not necessarily adjacent to the alternant. Further implications of this assumption for contexts will be examined in section 4.1.

3.2 String differences

The learning algorithm proposed here is based on the notion of string differences introduced by (Goldsmith 2011). This approach required defining an alphabet of symbols A and a binary concatenation operator $\bullet$ (also represented by simple juxtaposition). The alphabet is augmented by adding a null element for concatenation (indicated $\emptyset$) as well as inverse for each letter in A. The inverse of $a \in A$ is a^{-1}, and $aa^{-1} = a^{-1}a = \emptyset$. Moreover, $(ab)^{-1} = b^{-1}a^{-1}$. These definitions establish *group structure* over the set of all strings in the extended alphabet.

The *right difference* of strings *s* and *t* is defined as $\frac{s}{t}R = t^{-1} \bullet s$. Similarly, the *left difference* of *s* and *t* is $\frac{s}{t}L = s \bullet t^{-1}$. The following examples clarify this notation:

(12) $\frac{\textit{jumps}}{\textit{jumped}}R = (\textit{jumped})^{-1}\textit{jumps} =$
$(\textit{ed})^{-1}(\textit{jump})^{-1}\textit{jumps} = (\textit{ed})^{-1}s = \frac{s}{\textit{ed}}$

(13) $\frac{\textit{undo}}{\textit{redo}}L = \textit{undo}(\textit{redo})^{-1} =$
$\textit{undo}(\textit{do})^{-1}(\textit{re})^{-1} = \textit{un}(\textit{re})^{-1} = \frac{\textit{un}}{\textit{re}}$

For our purposes, it is sufficient to interpret string differences as ordered pairs of strings.[2] In its turn, *left/right commonality* can be defined as the longest common prefix/suffix of two strings. The left commonality of *jumps* and *jumped* is *jump*, and the right commonality of *undo* and *redo* is *do*.

Given a *paradigm* (set of strings) P with n elements, its left/right *self-difference array* is the $n \times n$ array D such that $D[i][j]$ is the left/right difference of $P[i]$ and $P[j]$. The array of commonalities is defined similarly. A paradigm is *regular* if each row in its self-difference array has a single common nominator and all elements in its commonality array are identical (ignoring the main diagonal).

	jump	jumps	jumped	jumping
jump		$\frac{\emptyset}{s}$	$\frac{\emptyset}{ed}$	$\frac{\emptyset}{ing}$
jumps	$\frac{s}{\emptyset}$		$\frac{s}{ed}$	$\frac{s}{ing}$
jumped	$\frac{ed}{\emptyset}$	$\frac{ed}{s}$		$\frac{ed}{ing}$
jumping	$\frac{ing}{\emptyset}$	$\frac{ing}{s}$	$\frac{ing}{ed}$	

Figure 1: Regular paradigm: right self-differences of {*jump, jumps, jumped, jumping*}

	try	tries	tried	trying
try		$\frac{y}{ies}$	$\frac{y}{ied}$	$\frac{\emptyset}{ing}$
tries	$\frac{ies}{y}$		$\frac{s}{d}$	$\frac{ies}{ying}$
tried	$\frac{ied}{y}$	$\frac{d}{s}$		$\frac{ied}{ying}$
trying	$\frac{ing}{\emptyset}$	$\frac{ying}{ies}$	$\frac{ying}{ied}$	

Figure 2: Non-regular paradigm: right self-differences of {*try, tries, tried, trying*}

This notion of self-difference is still limited to prefixes and suffixes. Let $P = \{w_1, ..., w_n\}$ be a paradigm whose left and right self-difference arrays are regular, with l and r denoting its (unique) left and right commonality respectively. Omitting some details for the sake of space, we define the set of *internal difference substrings* of P as $\{l^{-1}w_1r^{-1}, ..., l^{-1}w_nr^{-1}\}$.

Under the assumptions outlined previously, the task of identifying alternations reduces to finding segment-sized (or smaller) differences between realizations of the same morpheme. The following recursive definition captures this idea:

(14) An *alternation* is the set of internal difference substrings of a paradigm that is regular if any previously determined alternations are ignored and, moreover, satisfies two conditions:

(i) the paradigm's left or right commonality is a non-empty string;

(ii) none of the difference substrings are longer than one character.

3.3 A two-step algorithm

In the input, morphs are arranged into sets by gloss tag, each morph forming its own *group*. A fragment of the input is shown below.

(15) Q: {[mɤ], [me]}
PST: {[dɤ], [de], [tɤ], [te]}
P1PL: {[bez], [ebez], [ɤbɤz]}

The definition introduced in (14) lends itself naturally to a two-step iterative algorithm that calculates self-differences for each set of morphs, identifying alternations as it proceeds. The *extraction step* employs the definitions introduced above to find all possible alternations between groups within each set. The *reduction step* collapses all groups in each set that are identical up to known alternations, essentially factoring out some of the

[2] As Goldsmith (2011) points out, one can also think of the right difference of s and t as a *function* that maps t to s.

differences and making more alternations accessible to subsequent passes. The algorithm alternates between extraction and reduction until the number of groups stops decreasing.

Consider the toy example in (15). The first iteration starts out with no known alternations; the only morph set conforming to (14) is Q. The extraction step discovers one alternation: {e, ɤ}. The reduction step then collapses all morph groups that are identical up to this alternation:

(16) Q: $\left\{ \begin{bmatrix} \text{m ɤ,} \\ \text{m e} \end{bmatrix} \right\}$

PST: $\left\{ \begin{bmatrix} \text{d ɤ,} \\ \text{d e} \end{bmatrix}, \begin{bmatrix} \text{t ɤ,} \\ \text{t e} \end{bmatrix} \right\}$

P1PL: $\left\{ \begin{bmatrix} \text{b e z} \end{bmatrix}, \begin{bmatrix} \text{e b e z,} \\ \text{ɤ b ɤ z} \end{bmatrix} \right\}$

At the second iteration (17), members of the {e, ɤ} set are now treated as the same segment, and PST and P1PL satisfy the conditions of (14). They yield two new alternations, {d, t} and {∅, e, ɤ}, allowing the reduction step to collapse both PST and P1PL:

(17) Q: $\left\{ \begin{bmatrix} \text{m ɤ,} \\ \text{m e} \end{bmatrix} \right\}$

PST: $\left\{ \begin{bmatrix} \text{d ɤ,} \\ \text{d e,} \\ \text{t ɤ,} \\ \text{t e} \end{bmatrix} \right\}$

P1PL: $\left\{ \begin{bmatrix} \text{∅ b e z,} \\ \text{e b e z,} \\ \text{ɤ b ɤ z} \end{bmatrix} \right\}$

At this point no further reduction is possible, and the algorithm halts.

3.4 Intermediate results

The Mishar Tatar sample contained 55 different gloss tags and 160 surface realizations. The algorithm converged after three iterations, collapsing the morphs into 85 groups.

Correct	{d, n, t}, {d, t}, {∅, k, g}, {l, n}, {k, g}, {∅, e, ɤ}, {a, ä}, {e, ɤ}
Incomplete	{∅, ɤ}, {∅, ä}
Incorrect	{g, s}, {n, ŋ}

Table 2: Extracted alternations

The following output snippets illustrate the work of the reduction step. Multiple processes affecting the same morpheme can be learned (18); this also holds for alternations with zero (19). Not every set of morphs is reduced to a single group; the algorithm has successfully learned that the ATTR gloss corresponds to three different attributivizer suffixes, each of which has multiple realizations (20).

(18)
```
---- PL
---- Group 0
[['n' 'ä' 'r']
 ['n' 'a' 'r']
 ['l' 'ä' 'r']
 ['l' 'a' 'r']]
```

(19)
```
---- ORD
---- Group 0
[['e' 'n' 'č' 'e']
 [' ' 'n' 'č' 'e']
 ['ɤ' 'n' 'č' 'ɤ']]
```

(20)
```
---- ATTR
---- Group 0
[['s' 'ɤ' 'z']
 ['s' 'e' 'z']]
---- Group 1
[['l' 'e']
 ['l' 'ɤ']]
---- Group 2
[['g' 'e']
 ['g' 'ɤ']
 ['k' 'e']]
```

One interesting observation is related to the learner's ability to retain group boundaries if the groups appear to represent distinct morphemes. This has a practical potential for detecting inconsistencies in labelling – such as the COMP gloss tag being used for the complementizer *dip* and the comparative suffix *-r{aä}k* (21), when CMPR is expected for the latter (22).

(21)
```
---- COMP
---- Group 0
[['d' 'i' 'p']]
---- Group 1
[['r' 'ä' 'k']
 ['r' 'a' 'k']]
```

(22)
```
---- CMPR
---- Group 0
[['r' 'a' 'k']]
```

4 Learning contexts

4.1 Rules and locality

Above, we have introduced a method of detecting likely phonological processes and collecting them as sets of alternating segments. This section makes the next logical step and focuses on patterns governing the distribution of alternants.

A straightforward way to formalize this task and define its boundaries is grounded in formal language theory. Two classes of subregular languages are particularly relevant to the discussion of phonology. One of them is Strictly Local languages (McNaughton and Papert 1971; Rogers and Pullum 2011). Given an alphabet Σ, a Strictly k-Local (SL) grammar can be expressed as a set of strings in Σ of length at most k. The corresponding language is the set of all strings in Σ that do not contain any of the strings in the grammar. Tier-based Strictly Local grammars (TSL) (Heinz et al. 2011) are a generalization of SL grammars. A k-TSL grammar can also be defined as a set of illicit strings; however, they only apply to a certain subset, or *tier*, of elements in Σ, ignoring any intervening elements that do not belong to the subset.

How can this model be used to produce a practical representation of a phonological process? One way is to link each alternant occurring in a surface form to a set of *trigger segments*, indicating whether they occur to the left or right. Each alternation should also be associated with a set of transparent segments – non-tier elements in TSL terms. This is essentially a *bigram model*, encoding dependencies between pairs of elements, and has a clear counterpart in 2-TSL grammars. Given an alternation with n variants, the process of learning the rule boils down to determining the directionality and partitioning the set of segments into $n+1$ subsets of triggers (for each alternant) and transparent segments.

4.2 Mutual information

Mutual information (MI) is a measure of dependence between two random variables, or the reduction of uncertainty in one random variable through the other (Cover and Thomas 2012).

(23) $$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}$$

For the task at hand, it is convenient to think of MI as the expected value of *pointwise mutual information* (PMI). In its turn, PMI is an indication of how much the probability of a particular pair of events differs from what it is expected to be assuming independence (Bouma 2009). Intuitively, PMI measures correlation (positive or negative) between events.

(24) $$PMI(x;y) = \log_2 \frac{p(x,y)}{p(x)p(y)}$$

The PMI metric is naturally applicable to learning of vowel harmony. In our case, the algorithm is expected to learn the triggers and transparent segments for each process, which translates into calculating PMI values with respect to each specific alternation. Instead of the full set of bigrams in the corpus, the input for this procedure is the set of bigrams containing an alternant (member of the alternation in question) and a context segment.

A character bigram can be defined either locally, as a *substring* in a word, or nonlocally, as a *subsequence* (potentially non-contiguous pair), using left or right contexts of the alternant. These two parameters – *locality* and *directionality* – yield four different modes of collecting bigrams; see Table 3 for a concrete example.

	Left	Right
Local	šɤ	ɤŋ
Nonlocal	#ɤ, bɤ, aɤ, šɤ	ɤŋ, ɤ#

Table 3: Bigrams for {e, ɤ} in the word *bašɤŋ*

Consider the voicing alternation {d, t} and the harmonic pair {e, ɤ}. Both have triggers to the left of the target; the former is a local process, while the latter is nonlocal. Both alternations are present in the past tense suffix:

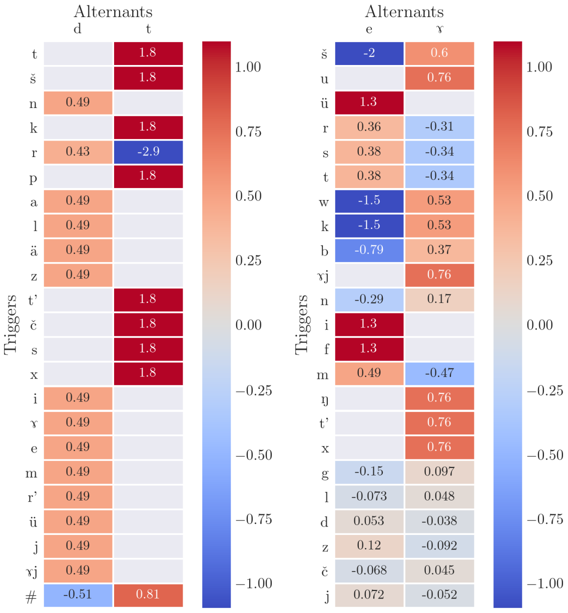
(25) a. ker-**de**
enter-PST

b. kɤčkɤr-**dɤ**
shout-PST

c. ɤrɤš-**tɤ**
scold-PST

d. teš-**te**
fall-PST

Presented in Figure 3 and Figure 4 are heat maps showing PMI values calculated for the alternations in question. High positive values in cells indicate attraction, whereas negative values correspond to elements repelling each other. Cells without values indicate unattested bigrams.

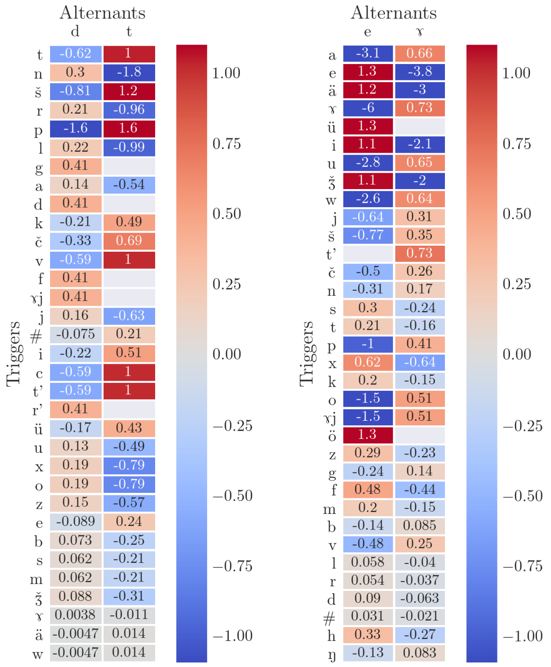


(a) Local process: {d, t}

(b) Vowel harmony: {e, ɤ}

Figure 3: PMI heat maps for local left bigrams; higher values indicate stronger attraction between segments

Local bigrams yield a very clear picture for the voicing alternation {d, t}. In Figure 3a, unexpected pairs – voiced trigger and unvoiced alternant, or vice versa – are either absent or have very low PMI values. However, local bigrams do not perform well on the vowel harmony pair {e, ɤ}. Figure 3b does indicate correct preference for some vowels, but the absolute values are comparably high for a number of consonants as well. With nonlocal bigrams the results are almost reversed. For {d, t} (Figure 4a), the pattern is obscured. However, nonlocal processes such as vowel harmony yield some correct information: for {e, ɤ} (Figure 4b) more vowels and fewer consonants exhibit strong positive or negative correlation tendencies.



(a) Local process: {d, t}

(b) Vowel harmony: {e, ɤ}

Figure 4: PMI heat maps for nonlocal left bigrams

The heat maps demonstrate that PMI values can be successfully used to match triggers to alternants. What is needed at this point is a procedure that would assign correct sets of transparent segments to each process – namely, the empty set for local processes and the set of consonants for vowel harmony.

Augmenting local bigrams with the notion of transparent segments produces a generalization applicable to both local and nonlocal processes. A left (right) local bigram consists of an alternant and the closest non-transparent segment to its left (right). One option, then, is to compare context segments directly in terms of how likely they are to be transparent for a given alternation. A non-transparent segment is expected to have high absolute PMI values – positive with the alternant it triggers and negative with all other alternants. The definition of MI (23) can be rewritten as follows:

(26) $$MI = \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) PMI(x;y)$$

Fixing x and normalizing the value by its probability to avoid unnecessarily high scores for rarely attested segments, we obtain the following metric:

(27) $$MI(x) = \sum_{y \in Y} p(x,y) PMI(x;y)$$

This allows to *rank* context segments by their MI value. The bigrams have to be calculated in the nonlocal mode to capture information about long-distance dependencies. Intuitively, the higher a segment is ranked, the more likely it is to be transparent with respect to the alternation in question. For each segment, the ranking also shows the alternant that corresponds to the highest PMI value. The ranking for {e, ɤ} (left contexts) is shown in (28).

(28)
```
---- Alternation: {'e', 'ɤ'}:
ŋ:    ɤ    0.00001362
h:    e    0.00001943
#:    e    0.00003585
d:    e    0.00004618
r:    e    0.00004704
...
z:    e    0.00034681
ö:    e    0.00042929
o:    ɤ    0.00051578
ɤj:   ɤ    0.00051578
k:    e    0.00053448
...
u:    ɤ    0.01335405
i:    e    0.01762960
ü:    e    0.01845926
ɤ:    ɤ    0.03614305
ä:    e    0.03741840
e:    e    0.04574272
a:    ɤ    0.04901854
```

As expected, most vowels have high values, whereas consonants tend to score low. Some vowels still end up in the middle – in particular, *o* and *ö*, which are uncharacteristic for this dialect and generally found in borrowed roots. Provided that the alternation set itself has been identified correctly, for every trigger segment the highest PMI value unerringly points at the correct alternant unless the segment is transparent.

4.3 Phonological viability and rule evaluation

Due to the limited data, the learner cannot be expected to have access to all possible contexts. Moreover, as shown in (28) above, the rankings of segments produced by calculating PMI tend to contain some degree of noise. It is at this point that phonological features come into play. Adopting the standard textbook definition, a *natural class* is a set of segments that share a particular value for some feature or a set of features (Odden 2013). A rule is considered *phonologically viable* just in case the sets of triggers of all alternants correspond to disjoint natural classes.[3][4]

Phonological viability introduces a straightforward way of producing generalizations. Combined with PMI rankings, it can be used to generate phonologically meaningful rules for known alternations. First, each trigger set is extended with segments in its natural class that have not occurred in the context of the given alternation. Second, any transparent segments that were accidentally added to the transparent list is removed from it if they are also found in one of the expanded trigger sets. These modifications produce *generalized rules*.

We use two metrics to evaluate and compare these rules. The primary objective is to explain as many instances of the given alternation as possible. This intuition is easy to formalize: an example is *explained* if it contains a correct trigger which is either adjacent to the alternant or separated only by transparent segments. Another option is to calculate the average PMI over all (segment, alternant) pairs, following the standard definition of mutual information shown in (24).

4.4 Assembling the pieces

In order to determine the best cutoff point in the ranking, each alternation A is processed as follows. At initialization, MI values are calculated once with nonlocal bigrams in order to rank the segments; all subsequent calculations are performed with local bigrams. The set of transparent segments, Transp_A, starts out empty. The algorithm traverses the ranking, starting with the lowest MI value. At each step, the selected segment is added to Transp_A, and both metrics (mutual information and explained examples) are recalculated. Thus, Transp_A is expanded incrementally until it contains all segments in the ranking; every step produces a new rule with triggers assigned according to the current PMI values. If the rule is phonologically viable, it is converted into a generalized rule, and the metrics are recalculated once more. The procedure is performed twice for each alternation, on left and right contexts separately. Once it halts, the best generalized rule is selected – which means that only phonologically viable configurations are eligible candidates.

The results are summarized in Table 4. For each alternation, MI and explained examples ratio (EE) are shown for the best rule based on left and right contexts. Colored rows indicate that the learner has both partitioned the set of attested context segments and determined whether the trigger is to the left or to the right correctly with respect to the ground truth.

Alternation	MI (left)	EE (left)	MI (right)	EE (right)
{d, n, t}	1.4277	**1.0000**	0.0000	0.0000
{d, t}	0.8079	**0.9864**	0.0421	0.3143
{∅, k, g}	0.4951	**0.4909**	0.0000	0.0000
{l, n}	0.5547	**0.9738**	0.0514	0.1154
{k, g}	0.0814	**0.1653**	0.0000	0.0000
{∅, ɤ, e}	0.8431	**0.6220**	0.6205	0.5357
{a, ä}	0.8319	**0.9733**	0.8331	0.8387
{e, ɤ}	0.8825	**0.9857**	0.7148	0.7912
{∅, ɤ}	0.8113	1.0000	1.0000	1.0000
{∅, ä}	0.1651	0.4688	0.5586	**1.0000**

Table 4: Rule evaluation for correct and incomplete alternations

As mentioned in section 2, some alternations are involved in multiple processes. For instance, {k, g} conflates two assimilation processes with different directionality, while {∅, ɤ, e} corresponds to a combination of a local and nonlocal processes. As expected, they have lower scores.

The algorithm yields promising results for both local and nonlocal processes, provided that the alternation set itself is non-problematic. The vowel harmony case is further illustrated by the final heat map showing all non-transparent segments in the best generalized rule for {e, ɤ} in Figure 5. In particular, compare Figure 5 to Figures 3b and 4b and the ranking in (28).

[3] This is a simplification, as one of the trigger sets may form an unnatural class that corresponds to the general case – a fact captured by the Elsewhere Condition (Kiparsky 1973). The definition of phonological viability implements the Elsewhere Condition to some degree, as no natural class requirement is imposed on the set of transparent segments.

[4] Under this definition, classes of segments specified by disjunction are generally unnatural. While languages have a tendency to favor natural classes (definable by feature conjunction) in their rules (Halle and Clements 1983), exploring more relaxed definitions for the purposes of determining acceptable rules is an interesting avenue of future work.

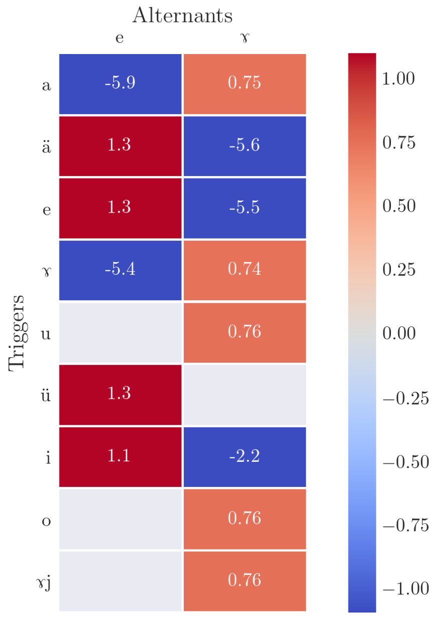


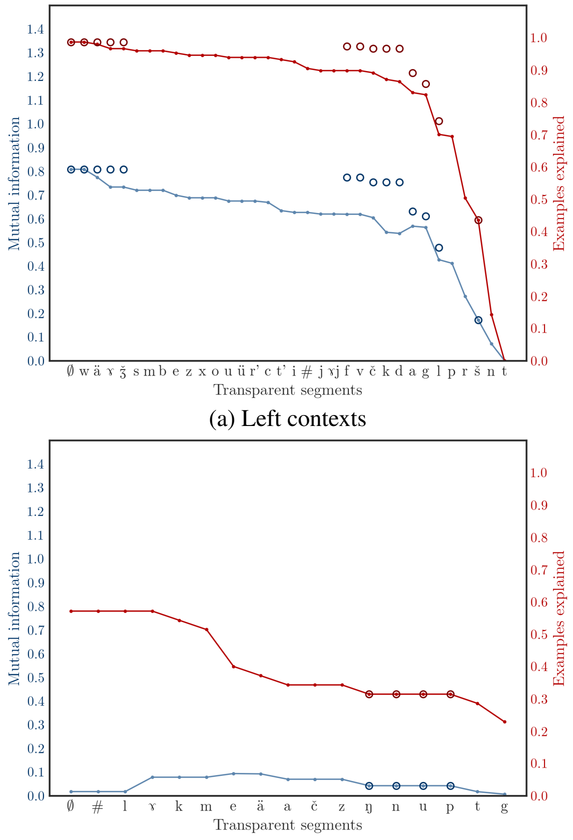
Figure 5: Final PMI heat map for vowel harmony: {e, ɣ}, left contexts

One way to gain insight into the procedure of context learning is to plot the step-by-step change of metric values depending on the set of transparent segments.

Figures 6–7 show graphs for left and right contexts with respect to the {t, d} and {e, ɣ} alternations. Each graph has context segments, ordered according to the MI ranking, along its x-axis. Each point corresponds to a step performed by the algorithm – or, equivalently, to a rule whose set of transparent segments contains all items on the x-axis up to and including that point. In addition, circle markers are present at every phonologically viable step and indicate values obtained by generalized rules.

Graphs for local and nonlocal processes display strikingly different behaviour. The typical picture for a local process is a monotonic sequence: both metrics start out high but decline steadily as more segments are declared transparent. For left-triggered processes, right contexts show noticeably lower values throughout the procedure – especially so if only phonologically viable steps are considered.

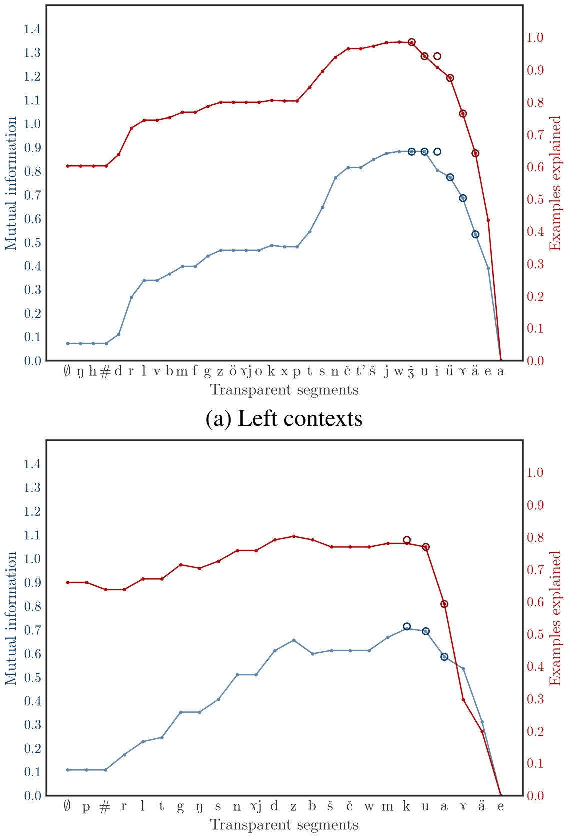
For vowel harmony (Figure 7) the plots start low and show a distinct peak once a sufficient number of segments are moved to the transparent set. The peak corresponds to the last consonant in the ranking.



(a) Left contexts

(b) Right contexts

Figure 6: MI and explained examples for {t, d}



(a) Left contexts

(b) Right contexts

Figure 7: MI and explained examples for {e, ɣ}

While the values for left contexts are still higher, the difference is not as great. This is an expected result: since vowels serve as both triggers and targets of harmony, most vowels in non-final syllables would have a harmonizing vowel both to the left and to the right.

5 Discussion

This paper presents an approach to learning (morpho)phonological phenomena from small annotated datasets that combines information-theoretic methods with linguistic information. The proposal includes an algorithm that discovers phonological alternations (represented as sets of segments) shared by multiple morphological paradigms. The notion of mutual information is used to assign a set of contexts to each alternant. Possible rules are then restricted to phonologically plausible configurations via a procedure reminiscent of regularization in machine learning. This approach is applicable to both local and nonlocal processes.

All these should be taken as interim results. One option for future work is to explore interaction between alternation sets. For example, it may be possible to decompose the complex {∅, e, ɤ} alternation by first factoring out the known vowel harmony pattern {e, ɤ}, leaving a simple local process. Other steps that follow directly from the results described here include predicting and reconstructing morphs that are absent from the dataset and, as a more practically oriented goal, identifying inaccuracies and instances of mislabelling in the data.

References

Diana Archangeli. 1988. Aspects of underspecification theory. *Phonology*, 5(2):183–207.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Jan Baudouin de Courtenay. 1876. Rez'ja i rez'jane. *Slavjanskij sbornik*, 3:223–371.

Thomas Cover and Joy Thomas. 2012. *Elements of information theory*. John Wiley & Sons.

Gallagher Flinn. 2014. Modeling neutrality in Mongolian vowel harmony. Manuscript.

John Goldsmith. 2011. A group structure for strings: Towards a learning algorithm for morphophonology. Technical report, Technical Report TR-2011-06, Department of Computer Science, University of Chicago.

John Goldsmith and Jason Riggle. 2012. Information theoretic approaches to phonological structure: the case of finnish vowel harmony. *Natural Language & Linguistic Theory*, 30(3):859–896.

Morris Halle and George N. Clements. 1983. *Problem book in phonology: a workbook for introductory courses in linguistics and in modern phonology*. MIT Press.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Paul Kiparsky. 1973. “Elsewhere” in phonology. In Paul. Kiparsky and Stephen R. Anderson, editors, *A festschrift for Morris Halle*, pages 93–106. New York: Holt, Rinehart & Winston.

Robert McNaughton and Seymour Papert. 1971. *Counter-Free Automata*. MIT Press.

MSU linguistic expedition. 1999–2012. Fieldwork materials. Lomonosov Moscow State University.

David Odden. 2013. *Introducing Phonology*. Cambridge University Press.

Frans Plank. 1998. The co-variation of phonology with morphology and syntax: A hopeful history. *Linguistic Typology*, 2:195–230.

James Rogers and Geoffrey Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Lili Szabó and Çagrı Çöltekin. 2013. A linear model for exploring types of vowel harmony. *Computational Linguistics in the Netherlands Journal*, 3:174–192.

David Allen Wax. 2014. *Automated grammar engineering for verbal morphology*. Ph.D. thesis, University of Washington.

Olga Zamaraeva. 2016. Inferring morphotactics from interlinear glossed text: combining clustering and precision grammars. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 141–150.