# A phoneme clustering algorithm based on the obligatory contour principle

**Mans Hulden**
Department of Linguistics
University of Colorado
mans.hulden@colorado.edu

## Abstract

This paper explores a divisive hierarchical clustering algorithm based on the well-known Obligatory Contour Principle in phonology. The purpose is twofold: to see if such an algorithm could be used for unsupervised classification of phonemes or graphemes in corpora, and to investigate whether this purported universal constraint really holds for several classes of phonological distinctive features. The algorithm achieves very high accuracies in an unsupervised setting of inferring a consonant-vowel distinction, and also has a strong tendency to detect coronal phonemes in an unsupervised fashion. Remaining classes, however, do not correspond as neatly to phonological distinctive feature splits. While the results offer only mixed support for a universal Obligatory Contour Principle, the algorithm can be very useful for many NLP tasks due to the high accuracy in revealing consonant/vowel/coronal distinctions.

## 1 Introduction[1]

It has long been noted in phonology that there seems to be a universal cross-linguistic tendency to avoid redundancy or repetition of similar speech features within a word or morpheme, especially if the phonemes are adjacent to one another. Many different names are given to variants of this general phenomenon in the linguistic literature: "identity avoidance" (Yip, 1998), "similar place avoidance" (Pozdniakov and Segerer, 2007), "obligatory contour principle" (OCP) (Leben, 1973), and "dissimilation" (Hempl, 1893). Some special cases such as haplology (avoidance of adjacent identical syllables) also fall in this general category of avoiding repetition along some dimension.

The general phenomenon itself is supported by robust, although inconsistent, evidence across a number of languages. An early example is the observation of Spitta-Bey (1880),[2] that the Arabic language tends to favor combination of consonant segments (phonemes) in morphemes that have different places of articulation; this was also later pointed out by Greenberg (1950) and those Semitic root outliers that deviate from this pattern were analyzed in depth in Frajzyngier (1979). In Proto-Indo-European (PIE) roots, which are mostly structured CVC, stop-V-stop combinations have been found to be statistically underrepresented (Iverson and Salmons, 1992). That is, PIE seems to obey a cross-linguistic constraint that disfavors two similar consonants in a root. Another specific example comes from Japanese, where the phenomenon called Lyman's law—which effectively says that a morpheme may consist of maximally one voiced obstruent—can also be interpreted as avoidance (Itô and Mester, 1986).

In light of such evidence, proposals have been put forth to define the concept of phoneme by distributional properties alone as opposed to the prevalent distinctive feature systems which are largely based on articulatory features (Fischer-Jørgensen, 1952). Elsewhere, after finding a statistical tendency to avoid similar place of articulation in word-initial and word-medial consonants, Pozdniakov and Segerer (2007) offer the argument

[1] All code data sets used are available at https://github.com/cvocp/cvocp

[2] Nun hat, wie schon längst bemerkt ist, die arabische Sprache die Neigung, solche Buchstaben in einem Worte zu vereinigen, deren Organe weit von einander entfernt liegen, wie Kehllaute und Dentale. Translation: Now, the Arabic language, as has long been noted, has the tendency to combine such letters in a word where the place of articulation is distant, such as gutturals and dentals (Spitta-Bey, 1880, p. 15).



*Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 290–300,
Vancouver, Canada, August 3 - August 4, 2017. 

that this phenomenon of "Similar Place Avoidance" is a statistical universal.

This phenomenon is often filed under the generic heading "obligatory contour principle" (Leben, 1973; McCarthy, 1986; Yip, 1988; Odden, 1988; Meyers, 1997; Pierrehumbert, 1993; Rose, 2000; Frisch, 2004). Originally, the OCP was applied as a theoretical constraint only to tone languages, with the argument that adjacent identical tones in *underlying forms* were rare, and this reflected an obligatory contour principle. The usage has since spread, and is assumed to account for segmental features other than tone.

It is unclear why the phenomenon is so widespread and why it manifests itself in the diverse ways it does. Accounts range from information compression to a diachronically visible hypercorrection by listeners who misperceive the signal and make the assumption that repetition is unlikely (Ohala, 1981).

This paper explores the simplest incarnation of the idea of similarity avoidance; namely, that two adjacent segments are preferably different in some way and that this difference reveals itself globally. That is, it is not assumed that the constraint is absolute; rather, an algorithm is developed that induces grouping of unknown phoneme symbols so as to maximize potential alternation of clusters in a sequence of symbols, i.e. a corpus. If the OCP holds for phonological or phonetic features—primarily places of articulation—such a clustering algorithm could group phonemes along the lines of distinctive features. While, as we shall see, the observations do not support the presence of a strong universal OCP effect, the top-level clusters discovered by the algorithm correspond nearly 100% to the distinction of consonants and vowels—or syllabic and non-syllabic elements if expressed in terms of features. Furthermore, a tier-based variant of the algorithm additionally groups consonants somewhat reliably into coronal/non-coronal places of articulation, and also often distinguishes front vowels from back vowels. This is true even if the algorithm is run on alphabetic representations. An evaluation of the ability to detect C/V distinction against a data set of 503 Bible translations (Kim and Snyder, 2013) is included, improving upon earlier work that attempts to distinguish between consonants and vowels in an unsupervised fashion (Kim and Snyder, 2013; Goldsmith and Xanthos, 2009; Moler and Morrison, 1983; Sukhotin, 1962). The algorithm is also more robust than earlier algorithms that perform consonant-vowel separation and works with less data, something that is also briefly evaluated.

This paper is structured as follows: an overview of previous work is given in section 2, mostly related to the simpler task of grouping consonants and vowels without labeled data, rather than identifying distinctive features. Following that, the general algorithm is developed in section 3, after which the experiments on both phonemic and graphemic representations in section 4 are reported. Four experiments are evaluated. The first uses phonemic data from 9 languages for clustering and evaluates clustering along distinctive feature lines. The second is a graphemic experiment that uses a data set of Bible translations in 503 languages where the task is to distinguish the vowels from the consonants; here, results are compared to Kim and Snyder (2013) on the same data set. That data is slightly noisy, motivating the third experiment, which is also graphemic and evaluates consonant-vowel distinctions on vetted word lists from data taken from the ACL SIGMORPHON shared task on morphological reinflection (Cotterell et al., 2016). The ability of a tier-based variant of the algorithm to separate coronals from non-coronals is evaluated in a fourth experiment where Universal Dependencies corpora (Nivre et al., 2017) are used.

The main results are presented in section 5. Given the high accuracy of the algorithm in C/V distinction with very little data and its consequent potential applicability to decipherment tasks, a small practical example application is evaluated which analyzes a fragment of text, a manuscript of only 54 characters.

## 2 Related Work

The statistical experiments of Andrey Markov (1913) on Alexander Pushkin's poem *Eugene Onegin* constitute what is probably one of the earliest discoveries of the fact that significant latent structure can be found by examining immediate co-occurrence of graphemes in text. Examining a 20,000-letter sample of the poem, Markov found a strong statistical bias that favored alternation of consonants and vowels. A number of computational approaches have since been investigated that attempt to reveal phonological structure in corpora. Often, orthography is used

as a proxy for phonology since textual data is easier to come by. A spectral method was introduced by Moler and Morrison (1983) with the explicit purpose of distinguishing consonants from vowels by a dimensionality reduction on a segment co-occurrence matrix through singular value decomposition (SVD). An almost identical SVD-based approach was later applied to phonological data by Goldsmith and Xanthos (2009). Hidden Markov Models coupled with the EM algorithm have also been used to learn consonant-vowel distinctions (Knight et al., 2006) as well as other latent structure, such as vowel harmony (Goldsmith and Xanthos, 2009). Kim and Snyder (2013) use Bayesian inference supported by simultaneous language clustering to infer C/V-distinctions in a large number of scripts simultaneously. We compare our results against a data set published in conjunction with that work. More directly related to the current work are Mayer et al. (2010) and Mayer and Rohrdantz (2013) who work with models for visualizing consonant co-occurrence in a corpus.

### 2.1 Sukhotin's algorithm

Sukhotin's algorithm (Sukhotin, 1962, 1973) is a well-known algorithm for separating consonants from vowels in orthographic data; good descriptions of the algorithm are given in Guy (1991) and Sassoon (1992). The idea is to start with the assumption that all segments in a corpus are consonants, then repeatedly and greedily find the segment that co-occurs most with other segments, and declare that a vowel. This is performed until a stopping condition is reached. The algorithm is known to perform surprisingly well (Foster, 1992; Goldsmith and Xanthos, 2009), although it is limited to the task it was designed to do—inferring a C/V-distinction (with applications to decipherment) without attempting to reveal any further structure in the segments. All the syllabic/nonsyllabic distinction results in the current work are compared with the performance of Sukhotin's algorithm.

## 3 General OCP-based algorithm

At the core of the new clustering algorithm is the OCP-observation alluded to above, already empirically established in (Markov, 1913, 2006), that there is a systematic bias toward alternating adjacent segments along some dimension. To reveal this alternation, one can assume that there is a natural grouping of all segments into two initial sets, called **0** and **1**, in such a way that the total number of **0-1** or **1-0** alternations between adjacent segments in a corpus is maximized. For example, consider a corpus of a single string **abc**. This can be split into two nonempty subsets in six different ways: **0** = {ab} and **1** = {c}; **0** = {a} and **1** = {bc}; **0** = {ac} and **1** = {b}, and their symmetric variants which are produced by swapping **0** and **1**. Out of these, the best assignment is **0** = {ac} and **1** = {b}, since if reflects an alternation of sets where **abc** $\mapsto$ **010**. The 'score' of this assignment is based on the number of adjacent alternations, in this case 2 (**01** and **10**).

Outside of such small examples which split perfectly into alternating sets, once this optimal division of all segments into **0** and **1** is found, there may remain some residue of adjacent segments in the same class (**0-0** and **1-1**). The sets **0** and **1** can then be partitioned anew into subsets **00**, **01** (from **0**) and **10** and **11** (from **1**). Again, there may be some residue, and the partitioning procedure can be applied recursively until no further splitting is possible, i.e. until all of the adjacent segments fall into different clusters in the hierarchy.

More formally, given a corpus of words $w_1, \ldots, w_n$ and where each word is a sequence of symbols $s_1, \ldots, s_m$, this top-level objective function that we want to maximize can be expressed as

$$\sum_w \sum_i \mathbb{1}(\textbf{Group}(s_i) \neq \textbf{Group}(s_{i+1})) \qquad (1)$$

where **Group(s)** is the set that segment $s$ is in.

Given a suggested split of all the segments in a corpus into, say, the top-level disjoint sets **0** and **1**, we obviously do not need to examine the whole corpus to establish the score but can do so by simply examining bigram counts of the corpus.

Still, finding just the top-level split of segments into **0** and **1** is computationally expensive if done by brute force by trying all the possible assignments of segments into **0** and **1** and evaluating the score for each assignment. Since there are $2^n$ ways of partitioning a set of segments into two subsets (ignoring the symmetry of **0** and **1**), such an approach is feasible in reasonable time only for small alphabets ($< 25$, roughly).

To address the computational search space problem, the algorithm is implemented by a type

of simulated annealing (Kirkpatrick et al., 1983; Černỳ, 1985) to quickly find the optimum. The algorithm for the top-level split proceeds as follows:

(1) Randomly divide the set $S$ into $S'$ and $S''$

(2) Draw an integer $p$ from *Uniform*(1...K), where $K$ depends on the cooling schedule

(3) Swap $p$ random segments between $S'$ and $S''$

(4) If score is higher after swap, keep swap else discard swap. Go to (2).

The idea is to begin with an arbitrary partition of $S$ into $S'$ and $S''$, then randomly trying successively smaller and smaller random swaps of segments between the two sets according to a cooling schedule, always keeping the swap if the score improves. The cooling schedule was tested against corpora that use smaller alphabets where the answer is known beforehand by a brute-force calculation. The cooling was made slow enough to give the correct answer in 100/100 tries on such development corpora. In practice, this yields an annealing schedule where early swaps (the size of $K$) are sometimes as large as $|S|$, ending in $K$ equaling 1 for several iterations before termination. This splitting is repeated recursively to produce new sub-splits until no splitting is possible, i.e. the score cannot improve by splitting a set into two subsets.

### 3.1 A tier-based variant

Many identity avoidance effects have been documented that seem to operate not by strict adjacency, but over intervening material, such as consonants and vowels, as discussed in the introduction. For example, Rose (2000) argues that OCP effects apply to adjacent consonants across intervening vowels in Semitic languages. This motivates a tier-based variant of the algorithm. In this modification, instead of repeatedly splitting sets based on a residue of adjacent segments that belong to the same set, we instead modify the corpus, removing segments after each split. Each time we split a set $S$ into $S'$ and $S''$ based on a corpus $C$, we also create new corpora $C'$ and $C''$ where segments in $S''$ are removed from $C'$ and segments in $S'$ are removed from $C''$. Splitting then resumes recursively for $S'$ and $S''$, where $S'$ uses the corpus $C'$ and $S''$ the corpus $C''$. Figure 1 shows an example of this. Here, the initial

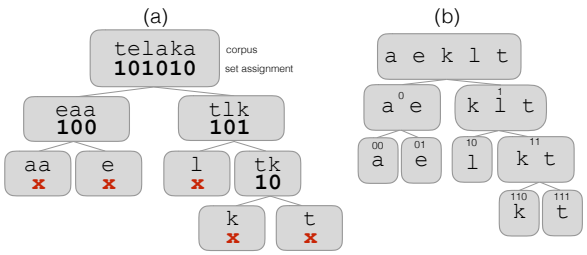


Figure 1: Illustration of the tier-based variant of the clustering algorithm. The left-hand side (a) shows the original corpus (the single word **telaka**), where each character is assigned a top-level grouping, after which the corpus is modified to remove characters in the respective sets **0** and **1**. The algorithm is then applied recursively to the modified corpora. The resulting clustering is shown in (b).

corpus $C$ = **telaka**, and the initial segment set $S = \{a, e, k, l, t\}$ is split into $S' = \{a, e\}$ and $S'' = \{k, l, t\}$ on a first iteration. Likewise, the corpus is now modified by removing the $S'$ and $S''$ segments from $C''$ and $C'$ respectively, yielding new corpora $C'$ = **eaa** and $C''$ = **tlk**, and splitting proceeds on these subcorpora. This way, if, say, consonants and vowels operate on different tiers and get split first into top-level sets, the remaining consonants will become adjacent to each other on the next iteration, as will the vowels.

## 4 Experiments

Four experiments are evaluated; the first experiment performs a full hierarchical clustering on phonemic data in 9 typologically divergent languages. The clusters are evaluated according to the following simple criterion: counting the number of splits in the tree that correspond to a split that could be expressed through a single phonological ± feature. For example, if the top level split in the tree produced corresponds to exactly the consonants and vowels, it is counted as a 1, since this corresponds to the partitioning that would be produced by the phonological feature [±syllabic]. If there is no way to express the split through a single distinctive feature, it is counted as a 0. A standard phonological feature set like that given in sources such as Hayes (2011) or PHOIBLE (Moran et al., 2014) is assumed. As mentioned above, the hypothesis under examination is that if the OCP is a strong universal principle, some non-significant number of subclusters coinciding with single phonological distinctive features should be

| Language | Source | Sample |
|---|---|---|
| Arapaho | (Cowell and Moss Sr, 2008) | towohei hiiθeti? tohnooke? toothei?eihoo . . . |
| Basque | Wikipedia + g2p | meʃikoko iriburuko espetʃe batean sartu zuten eta meʃiko . . . |
| English | (Brent and Cartwright, 1996) | ju want tu si ðə bʊk lʊk ðɛrz ə bɔɪ wɪð hɪz hæt . . . |
| Finnish | (Aho, 1884) + g2p | vai oli eilen kolmekymmentæ kotoapæinkø se matti ajelee . . . |
| Hawaiian | Wikipedia + g2p | ʔo ka ʔōlelo hawaiʔi ka ʔōlelo makuahine a ka poʔe maoli . . . |
| Hungarian | (Gervain and Erra, 2012) | idʒ nintʃ jɒj dɛ tʃɛtʃɛ hol ɒ montʃikɒ hol vɒn ɒ montʃi itt ɒ . . . |
| Italian | Wikipedia + g2p | tʃitta eterna kon abitanti e il komune piu popoloso ditalia . . . |
| Polish | (Boruta and Jastrzebska, 2012) | gd̑zie jest bartuc̣ gd̑zie jest ɲe ma xod̑ tu a kuku tso xovaʃ . . . |
| Spanish | (Taulé et al., 2008) + g2p | un akuerdo entre la patronal i los sindikatos franθeses sobre . . . |

Table 1: The data used for the phonemic clustering experiment, with sources indicated and a sample.

found. Both the non-tier algorithm and the tier-based algorithm is evaluated.

In the second experiment, the capacity of the algorithm to distinguish between consonants and vowels is evaluated, this time with graphemic data. To separate consonants from vowels—the most significant dimension of alternation between adjacent segments—the algorithm is run only for the top-level split, and it is assumed that the top two subsets will represent the consonants and vowels. Here, the results are compared with those of Kim and Snyder (2013), who train a hierarchical Bayesian model to perform this distinction over all the 503 languages at the same time. Sukhotin's algorithm is also used as another baseline.

In the third experiment, the capacity to distinguish consonants and vowels in graphemic data in the form of word lists—i.e. where no frequency data is known—is evaluated compared against Sukhotin's algorithm.

### 4.1 Phonemic splitting

Nine languages from a diverse set of sources were used for this experiment (see Table 1). Some of the language data were already represented as phonemes (English, Hungarian, and Polish), while for the others, which have close-to-phonemic writing systems, a number of grapheme-to-phoneme (g2p) rules were created manually to convert the data into an International Phonetic Alphabet (IPA) representation. The conversion was on the level of the phoneme—actual allophones (such as /n/ being velarized to [ŋ] before /k/ in most languages or /d/ being pronounced [ð] intervocalically in Spanish) were not modeled. Table 1 summarizes the data and gives a sample of each corpus.

For this data, the clustering algorithm was run as described above and each split was annotated

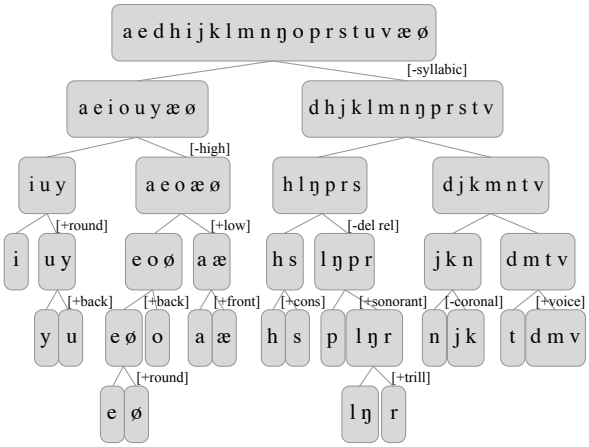


Figure 2: Resulting Finnish clusters with manual annotation of the distinctive feature splits.

with information about whether the split *could* be defined in terms of a single distinctive feature. Figure 2 shows the output of such a tree produced by the algorithm, with manual feature annotations.

The percentage of correctly identified top-level splits (which are syllabic/non-syllabic segments) is also given, together with the corresponding results from Sukhotin's C/V-inference algorithm, and Moler & Morrison's SVD-based algorithm.

### 4.2 C/V distinction in Bible translations

This experiment relies on word lists and frequency counts from Bible translations covering 503 distinct languages. Of these, 476 use a Latin alphabet, 26 a Cyrillic alphabet, and one uses Greek. The data covers a large number of language groups, and has been used before by Kim and Snyder (2013) to evaluate accuracy in unsupervised C/V-distinction.

The algorithms were evaluated in two different ways: one, on a task where each C and V set is inferred separately for each language, and two, in

a task where all languages' consonants and vowels are learned at once, as if the corpus were one language, for clearer comparison with earlier work. Both token-level accuracy and type-level accuracy are given, again, for comparability reasons. For this data set, Sukhotin's C/V-algorithm and Moler & Morrison's algorithm were used as baselines in addition to the results of Kim and Snyder (2013).

### 4.3 C/V-distinction with word lists

An additional experiment evaluates the algorithm's capacity to perform C/V-distinction against Sukhotin's algorithm on a data set of 10 morphologically complex languages where lists of inflected forms were taken from the ACL SIGMORPHON shared task data (Cotterell et al., 2016). In this case, we have no knowledge of the frequency of the forms given, but need to rely only on type information. The Arabic data was transliterated into a latinate alphabet (by DIN 31635), with vowels marked. For the other languages, the native alphabet was used. Per-type accuracy is reported.

## 5 Results

On the first task, which uses phonemic data, consonant/vowel distinction accuracy is 100% throughout (see Table 2). Sukhotin's algorithm also performs very well in all except two languages. English, in particular, is a surprising outlier, with Sukhotin's algorithm only classifying 21.62% correctly. This is probably due to there existing a proportionately large number of syllabic phonemes in English (13/37). Moler & Morrison's algorithm has less than perfect accuracy in three languages. There is great variation in the OCP algorithm's capacity to produce splits that coincide with phonological features in both the tier-based and non-tier variants. Roughly speaking, the larger the phoneme inventory, the less likely it is for the splits to align themselves in accordance with phonological features. Also, since the tier-based variant naturally leads to more splits, the figures appear higher since splits in lower levels of the tree, which contain few phonemes, can almost always be done along distinctive feature lines. The depth of the induced tree also correlates with the variety of syllable types permitted in the language. An extreme example of this is Hawaiian (Figure 3), which only permits **V** and **CV** syllables, yielding a very shallow tree where no consonants are split beyond the first level. English and Polish lie

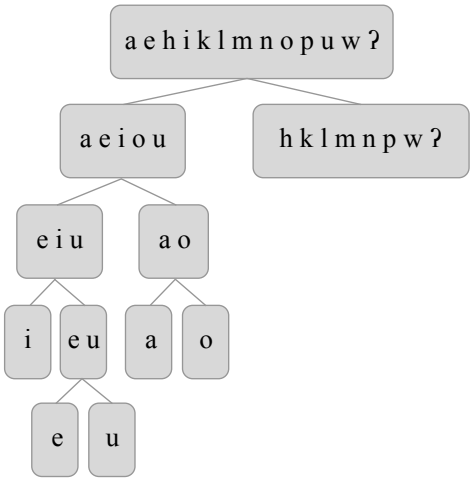


Figure 3: Hawaiian clusters reveal a predominantly CV/V syllable type since the non-syllabic branch of the tree is shallow.

at the other extreme, with 37 splits each. This circumstance may perhaps be further leveraged to infer syllable types from unknown scripts.

On the C/V inference task for 503 languages, the OCP algorithm outperforms Sukhotin's algorithm and Kim and Snyder (2013) (K&S) when each language is inspected individually (see Figure 3). However, for the case where we learn all distinctions at once, the OCP algorithm produces an identical result with Sukhotin. Here the token level accuracy also exceeds K&S with 99.89 vs. 98.55.

The already high accuracy rate of the OCP algorithm on the Bible translation data is probably in reality even higher, especially when all languages are inspected at the same time. Out of the 343 grapheme types, OCP and Sukhotin only misclassify 7, and upon closer manual inspection, it is found that only two of these are bona fide errors. Five are errors in the gold standard—all in the Cyrillic-based data (see Table 5 for an overview of the errors in the gold standard or the classifications). The first actual error, Cyrillic **s**, only occurs in five word types in the entire corpus, and is always surrounded by other consonants. The other error, **ŏ**, is more difficult to interpret—it occurs in three typologically different languages: Akoose (bss), Northern Grebo (gbo), and Peñoles Mixtec (mil).

On the third task, where only word lists are available from grapheme classification into C/V, the OCP algorithm performs equally to Sukhotin's algorithm, except for one language (Navajo),

| Language | Splits OCP | | Splits OCP(tier) | | C/V (OCP) | C/V (Sukh.) | C/V (M&M) | Inventory size |
|---|---|---|---|---|---|---|---|---|
| Arapaho | 9/14 | (62.29) | 11/15 | (73.34) | 100.0 | 100.0 | 100.0 | 16 |
| Basque | 8/14 | (57.14) | 16/20 | (80.00) | 100.0 | 100.0 | 100.0 | 21 |
| English | 3/12 | (25.00) | 15/25 | (60.00) | **100.0** | 21.62 | 94.59 | 37 |
| Finnish | 14/16 | (87.50) | 17/19 | (89.47) | 100.0 | 100.0 | 100.0 | 20 |
| Hawaiian | 4/5 | (80.00) | 8/12 | (66.67) | **100.0** | **100.0** | 92.30 | 13 |
| Hungarian | 10/20 | (50.00) | 21/31 | (67.74) | **100.0** | 96.97 | **100.0** | 33 |
| Italian | 7/11 | (63.64) | 15/20 | (75.00) | 100.0 | 100.0 | 100.0 | 22 |
| Polish | 10/21 | (47.61) | 23/33 | (69.70) | **100.0** | **100.0** | 97.30 | 37 |
| Spanish | 10/15 | (66.67) | 16/21 | (76.19) | 100.0 | 100.0 | 100.0 | 22 |

Table 2: Phonemic data: fraction of cluster splits that go exactly along single distinctive features (Splits with OCP/OCP (tier)), together with percentage. Also given are C/V-distinction accuracy (per type) for the OCP algorithm (OCP), Sukhotin's algorithm (Sukh.), Moler and Morrison's algorithm (M&M).

| | | OCP | Sukhotin | M&M | K&S |
|---|---|---|---|---|---|
| Individual | Type | **95.10** | 92.50 | 94.15 | – |
| | Token | **96.55** | 93.65 | 95.59 | 95.99 |
| All | Type | **96.43** | **96.43** | 89.79 | – |
| | Token | **99.89** | **99.89** | 99.79 | 98.55 |

Table 3: Results on the 503-language Bible translations on consonant-vowel distinction. Both type and token accuracy are included. The *Individual* column shows the macro-averaged results on running all languages individually, and the *All*-column shows the results of running all data at once. Here, 'OCP' is the current algorithm; 'Sukhotin' is Sukhotin's algorithm, 'M&M' is the SVD-method in Moler & Morrison (1983), and 'K&S' is the method given in Kim & Snyder (2013).

where the OCP algorithm misclassifies one symbol less (see Figure 4).

## 6 Application to text fragments: the arrow of the gods

Given that the algorithm performs very well on consonant-vowel distinctions and groups segments along distinctive features better with small alphabets, an additional experiment was performed on a small manuscript to get a glimpse of potential application to cryptography and the decipherment of substitution ciphers. In this experiment, the writing system is known to be alphabetic (in fact Cyrillic), and the purpose is to examine the clustering induced by so little available data.

| Language | OCP | Sukhotin | M&M |
|---|---|---|---|
| Arabic | 1/40 (97.50) | 1/40 (97.50) | 1/40 (97.50) |
| Finnish | 0/31 (100.0) | 0/31 (100.0) | 0/31 (100.0) |
| Georgian | 1/33 (96.97) | 1/33 (96.97) | **0/33 (100.0)** |
| German | **1/30 (96.67)** | **1/30 (96.67)** | 2/30 (93.33) |
| Hungarian | 1/33 (96.97) | 1/33 (96.97) | 1/33 (96.97) |
| Maltese | 2/30 (93.33) | 2/30 (93.33) | **0/30 (100.0)** |
| Navajo | 2/30 (93.33) | 3/30 (90.00) | **1/30 (96.67)** |
| Russian | **0/34 (100.0)** | **0/34 (100.0)** | 2/34 (94.12) |
| Spanish | **0/33 (100.0)** | **0/33 (100.0)** | 2/33 (93.94) |
| Turkish | 0/34 (100.0) | 0/34 (100.0) | 0/34 (100.0) |
| Average | **97.48** | 97.14 | 97.25 |

Table 4: Per type accuracy on C/V-distinction on word lists. Listed are the number of misclassifications, and the accuracy per type.

The birch bark letter number 292 found in 1957 in excavations in Novgorod, Russia, is the oldest known document in a Finnic language (Karelian), stemming most likely from the early 13th century (Haavio, 1964). The document consists of only 54 symbols, written in Cyrillic.[3] The clustering method (see Figure 4) identifies the vowels and consonants, except for the grapheme **y** (/u/). This is probably because the short manuscript renders the word **nuoli** (Latinized form) 'arrow' inconsistently in three different ways, with Cyrillic **y** = /u/ occurring in different places, making the segment difficult for the algorithm. The high vowels /i/ and

[3]The exact translation of the contents is a matter of dispute; the first translation given by Yuri Yeliseyev in 1959 reads as follows (Haavio, 1964): God's arrow ten [is] your name // This arrow is God's own // [The] God directs judgment.

| Symbol | Class | Comments |
|---|---|---|
| ѕ | V | Macedonian, only occurs four times. |
| ъ | V | Cyrillic soft sign (neither vowel not consonant). |
| ө | V | Cyrillic; error, should be CYRILLIC SMALL LETTER BARRED O, a vowel. |
| ӏ | V | Halh Mongolian, incorrect words in corpus. |
| ӣ | C | Cyrillic, corresponds to the palatal approximant /j/, incorrect in gold. |
| ї | C | Ukrainian iotated vowel sounds /ji/, unclear if vowel or consonant. |
| ŏ | C | Bantu languages: high tone/long vowel in Bantu languages. |

Table 5: The only misclassified segments in the 503-Bible test. The column *Class* gives this 'incorrect' classification of the OCP algorithm. Most of these are errors in the data/gold standard. Only the Cyrillic **s** which occurs four times in the data (always adjacent to other consonants) and the **ŏ**-symbol are actually incorrect.

/u/ (left) are also separated from the non-high vowels (right) /a/, /o/, and /e/ (the Cyrillic soft sign also falls in this group). Sukhotin's algorithm, which only infers the consonants and vowels, makes one more mistake than the current algorithm.

## 7 Identifying coronal segments with the tier-based variant

Although the only really robust pattern reliably discovered by the algorithm is the distinction between consonants and vowels, there are strong patterns within some of the clusters that appear to be cross-linguistically constant, specifically with the tier-based variant. The first is that, whenever a five-vowel system is present (such as in Basque, Spanish, and Italian), after the topmost split which divides up the vowels and the consonants, the first split within the vowel group is almost always $\{a, o, u\}$ and $\{e, i\}$. A second pattern concerns coronal segments. The first split within the consonant group tends to divide the segments into coronal/non-coronal segments. This is not an absolute trend, but happens far above chance. This is also true when running the algorithm on graphemic data, where coronals can be identified. Table 6 gives an overview of how cross-linguistically coherent the resulting first consonant splits are. The data set is a selection of 14 languages from the Universal Dependencies 2.0 data (Nivre et al., 2017).

| Language | Second Consonant Group | #C |
|---|---|---|
| **Basque** | (c) l n (ñ) r s x z | 21 |
| **Catalan** | l n r s x z | 22 |
| **Irish** | d l n r s | 13 |
| **Dutch** | h l n r x z | 19 |
| **Estonian** | h l n r s | 16 |
| **Finnish** | h l n r s (š) (x) (z) | 21 |
| **German** | j l n r s x z | 21 |
| **Indonesian** | l n r s z | 20 |
| **Italian** | h l n r s (y) | 21 |
| **Latin** | d h l n r s | 16 |
| **Latvian** | č j ķ l ļ n ņ r s z ž | 24 |
| **Lithuanian** | j l n r s š z ž | 19 |
| **Portuguese** | ç j l n (ñ) r s x | 24 |
| **Slovak** | c ď j l ĺ n ň r s š z ž | 26 |

Table 6: The second consonant grouping found using the tier-based OCP algorithm. This is the split below the top-level consonant/vowel split. The characters in this set largely correspond to coronal sounds. The data comes from 14 languages in the Universal Dependencies 2.0 data set. Shown in parentheses are symbols outside the native orthography of the language (most likely from named entities and borrowings found in the corpora). The rightmost column shows the total number of identified consonants in the language. In particular, **l**, **n**, and **r** are always in this set, while **s** is nearly always present.

## 8 Conclusion & future work

This paper has reported on a simple algorithm that rests on the assumption that languages tend to exhibit hierarchical alternation in adjacent phonemes. While such alternation does not always occur for any individual adjacent segment pair, on

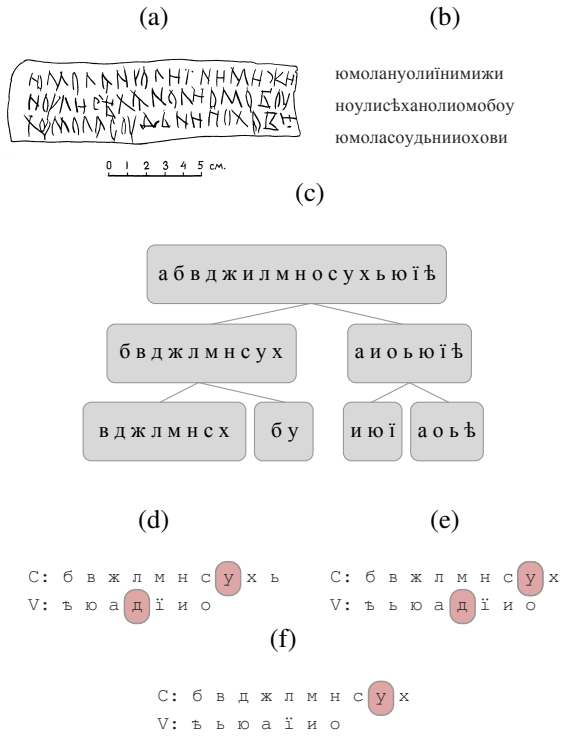


Figure 4: Clustering the graphemes in the 54-symbol *birch bark letter 292* manuscript (a), with transcription given in (b), and the results of OCP clustering (c). Also given are the C/V classifications produced by the Moler and Morrison (1983) algorithm (d), Sukhotin's algorithm (e), and the OCP algorithm (f), with errors marked with red boxes.

the corpus level this alternation largely holds and serves to reveal interesting structure in phonological organization. The top cluster discovered by the algorithm is also a highly reliable indicator of syllabic vs. non-syllabic segments, i.e. consonants and vowels, and improves upon the state-of-the-art in this unsupervised task. Interestingly, Sukhotin's C/V algorithm, which has similar performance (Sukhotin, 1962), can be interpreted as a greedy approximation of the first iteration in the current algorithm. A tier-based variant of the algorithm tends to detect front/back vowel contrasts and coronal/non-coronal contrasts as well, although this is more of a robust trend rather than an absolute.

Lower levels in the clustering approach are less reliable indicators of classical feature alternation, but can serve effectively to reveal aspects of syllable structure. For example, it is obvious from the Hawaiian clustering that the predominant syllable in the language is CV. One is led to conclude that the obligatory contour principle may be manifest in larger classes of segments (such as [±syllabic]), but not necessarily in on the fine-grained level. Some resulting cluster splits such as for example {m,p} vs. {b,f,t} (example from Basque) are often not only inseparable by a single feature split, but are not separable by any combination of features. This lack of evidence for a strong OCP may be in line with the vigorous debate in the phonological literature on the universal role of the OCP (see e.g. McCarthy (1986); Odden (1988)). Some languages (such as Finnish and Hawaiian) yield splits that almost always coincide with a single phonological feature, whereas other languages do not. Smaller inventories typically yield more robust results, although this may be partly due to chance factors—there are more ways to split a small set according to distinctive features than large sets.

Of interest is the utility of the extracted clusters in various supervised and semi-supervised NLP applications. For example, in algorithms that learn to inflect words from annotated examples (Ahlberg et al., 2015; Cotterell et al., 2016), it is often useful to have a subdivision of the segments that alternate, since this allows one to generalize behavior of classes of segments or graphemes, similar to the way e.g. Brown clusters (Brown et al., 1992) generalize over classes of words. Labeling segments with the position in a clustering tree and using that as a feature, for instance, is a cheap and straightforward way to inject this kind of knowledge into supervised systems designed to operate over many languages.

## Acknowledgements

Thanks to Andy Cowell for help with the Arapaho and Hawaiian datasets, Mike Hammond and Miikka Silfverberg for comments on an earlier version of this paper, and Francis Tyers for sharing his knowledge of Cyrillic writing systems and comments regarding the error analysis. Thanks also to Zygmunt Frajzyngier and Sharon Rose for general OCP-related discussion and comments. Several anonymous reviewers raised helpful points. This work has been partly sponsored by DARPA I20 in the program Low Resource Languages for Emergent Incidents (LORELEI) issued by DARPA/I20 under Contract No. HR0011-15-C-0113.

## References


Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics, Denver, Colorado, pages 1024–1029. http://www.aclweb.org/anthology/N15-1107.

Juhani Aho. 1884. *Rautatie [The Railroad]*. Werner-Söderström, Porvoo, Finland.

Luc Boruta and Justyna Jastrzebska. 2012. A phonemic corpus of Polish child-directed speech. In *Proceedings of the eighth international conference on Language Resources and Evaluation (LREC)*.

Michael R. Brent and Timothy A. Cartwright. 1996. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition* 61(1):93–125.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics* 18(4):467–479.

Vladimír Černỳ. 1985. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45(1):41–51.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*. Association for Computational Linguistics, Berlin, Germany.

Andrew Cowell and Alonzo Moss Sr. 2008. *The Arapaho Language*. University Press of Colorado.

Eli Fischer-Jørgensen. 1952. On the definition of phoneme categories on a distributional basis. *Acta linguistica* 7(1-2):8–39.

Caxton C. Foster. 1992. A comparison of vowel identification methods. *Cryptologia* 16(3):282–286.

Zygmunt Frajzyngier. 1979. Notes on the $R_1R_2R_2$ stems in Semitic. *Journal of Semitic Studies* 24(1):1–12.

Stefan A. Frisch. 2004. Language processing and segmental OCP effects. In Bruce Hayes, Robert Martin Kirchner, and Donca Steriade, editors, *Phonetically Based Phonology*, Cambridge University Press, pages 346–371.

Judit Gervain and Ramón Guevara Erra. 2012. The statistical signature of morphosyntax: A study of Hungarian and Italian infant-directed speech. *Cognition* 125(2):263–287.

John Goldsmith and Aris Xanthos. 2009. Learning phonological categories. *Language* 85(1):4–38.

Joseph H. Greenberg. 1950. The patterning of root morphemes in Semitic. *Word* 6(2):162–181.

Jacques B. M. Guy. 1991. Vowel identification: an old (but good) algorithm. *Cryptologia* 15(3):258–262.

Martti Haavio. 1964. The oldest source of Finnish mythology: Birchbark letter no. 292. *Journal of the Folklore Institute* 1(1/2):45–66.

Bruce Hayes. 2011. *Introductory Phonology*. John Wiley & Sons.

George Hempl. 1893. Loss of r in English through dissimilation. *Dialect Notes* (1):279–281.

Junko Itô and Ralf-Armin Mester. 1986. The phonology of voicing in Japanese: Theoretical consequences for morphological accessibility. *Linguistic Inquiry* pages 49–73.

Gregory K. Iverson and Joseph C. Salmons. 1992. The phonology of the Proto-Indo-European root structure constraints. *Lingua* 87(4):293–320.

Young-Bum Kim and Benjamin Snyder. 2013. Unsupervised consonant-vowel prediction over hundreds of languages. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1527–1536.

S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220(4598):671–680.

Kevin Knight, Anish Nair, Nishit Rathod, and Kenji Yamada. 2006. Unsupervised analysis for decipherment problems. In *Proceedings of the COLING/ACL*. Association for Computational Linguistics, pages 499–506.

William Ronald Leben. 1973. *Suprasegmental Phonology*. Ph.D. thesis, Massachusetts Institute of Technology.

A. A. Markov. 1913. Primer statisticheskogo issledovaniya nad tekstom "Evgeniya Onegina", illyustriruyuschij svyaz ispytanij v cep. *Izvestiya Akademii Nauk* Ser. 6(3):153–162.

A. A. Markov. 2006. An example of statistical investigation of the text "Eugene Onegin" concerning the connection of samples in chains. *Science in Context* 19(4):591–600.

Thomas Mayer and Christian Rohrdantz. 2013. PhonMatrix: Visualizing co-occurrence constraints of sounds. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Sofia, Bulgaria, pages 73–78.

Thomas Mayer, Christian Rohrdantz, Frans Plank, Peter Bak, Miriam Butt, and Daniel A Keim. 2010. Consonant co-occurrence in stems across languages: Automatic analysis and visualization of a phonotactic constraint. In *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground*. Association for Computational Linguistics, pages 70–78.

John J. McCarthy. 1986. OCP effects: Gemination and antigemination. *Linguistic Inquiry* 17(2):207–263.

Scott Meyers. 1997. OCP effects in optimality theory. *Natural Language & Linguistic Theory* 15(4):847–892.

Cleve Moler and Donald Morrison. 1983. Singular value analysis of cryptograms. *American Mathematical Monthly* pages 78–87.

Steven Moran, Daniel McCloy, and Richard Wright, editors. 2014. *PHOIBLE Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig. http://phoible.org/.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, et al. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

David Odden. 1988. Anti antigemination and the OCP. *Linguistic Inquiry* 19(3):451–475.

John Ohala. 1981. The listener as a source of sound change. In Carrie S. Masek, Roberta A. Hendrick, and Mary Frances Miller, editors, *Papers from the Parasession on Language and Behavior*, Chicago Linguistic Society, pages 178–203.

Janet Pierrehumbert. 1993. Dissimilarity in the Arabic verbal roots. In *Proceedings of NELS*. volume 23, pages 367–381.

Konstantin Pozdniakov and Guillaume Segerer. 2007. Similar place avoidance: A statistical universal. *Linguistic Typology* 11(2):307–348.

Sharon Rose. 2000. Rethinking geminates, long-distance geminates, and the OCP. *Linguistic Inquiry* 31(1):85–122.

George T. Sassoon. 1992. The application of Sukhotin's algorithm to certain non-English languages. *Cryptologia* 16(2):165–173.

Wilhelm Spitta-Bey. 1880. *Grammatik des arabischen Vulgärdialectes von Aegypten*. Hinrichs, Leipzig.

Boris V. Sukhotin. 1962. Eksperimental'noe vydelenie klassov bukv s pomoshch'ju EVM. *Problemy strukturnoj lingvistiki* pages 198–206.

Boris V. Sukhotin. 1973. Méthode de déchiffrage, outil de recherche en linguistique. *T. A. Informations* pages 1–43.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the sixth international conference on Language Resources and Evaluation (LREC)*.

Moira Yip. 1988. The obligatory contour principle and phonological rules: A loss of identity. *Linguistic Inquiry* 19(1):65–100.

Moira Yip. 1998. Identity avoidance in phonology and morphology. In Steven Lapointe, Diane Brentari, and Patrick Farrell, editors, *Morphology and its Relation to Phonology and Syntax*, CSLI, Stanford, pages 216–246.