An algorithm for learning phonological classes from distributional similarity

Connor Mayer

Abstract

An outstanding question in phonology is to what degree the learner uses distributional information rather than substantive properties of speech sounds when learning phonological structure. This paper presents an algorithm that learns phonological classes from only distributional information: the contexts in which sounds occur. The input is a segmental corpus, and the output is a set of phonological classes. The algorithm is first tested on an artificial language with both overlapping and nested classes reflected in the distribution. It retrieves the expected classes, and performs well as distributional noise is added. It is then tested on four natural languages. It distinguishes between consonants and vowels in all cases, and finds more detailed, language-specific structure. These results improve on past approaches, and are encouraging given the paucity of the input. Further refined models may provide additional insight into which phonological classes are apparent in the distributions of sounds in natural languages.

1 Introduction

An outstanding question in all areas of linguistics is how much of human language is innate and how much is learned from data. From this perspective, the question of how much information about phonological categories can be retrieved strictly from distributional information is of considerable interest to the field of phonology.

One of the central observations of phonological theory is that speech sounds tend to pattern according to phonetic similarity, both within and across languages. For example, processes like final obstruent devoicing, where voiced obstruents become voiceless word-finally or before voiceless consonants, are common across languages. This process often targets all voiced stops, affricates, and fricatives in a language. Despite the differences in place and manner of articulation across these sounds, they share two phonetic properties that cause them to be treated as a single class of sounds: near or complete impediment of the airflow out of the vocal tract, and vocal fold vibration.

Based on this robust typological generalization, classic work has suggested that there is a universal tendency for language learners to group sounds based on their phonetic properties

*This research was supported by the Social Sciences and Humanities Research Council of Canada. I would like to thank Bruce Hayes, Kie Zuraw, Yizhou Sun, Tim Hunter, Robert Daland, and Pat Keating for their guidance and support throughout this project. Thanks also to the attendees of the UCLA phonology seminar and the 2018 Annual Meeting on Phonology for their valuable questions and insights.

(e.g. Chomsky & Halle, 1968). Languages may use classes differently in their phonologies, but in principle the set of classes available across languages should be the same by virtue of shared human physiology.

There is evidence, however, that there are classes that do not appear to be phonetically coherent, such as the notorious Sanskrit "ruki" class (e.g. Kiparsky, 1973; Vennemann, 1974) or the triggers for Philadelphia /æ/-tensing (Labov et al., 2006). Mielke (2008) presents many such cases. Instances of variable patterning of a segment across languages also bear on this issue. For example, /l/ varies in whether a language treats it as [+continuant] or [−continuant] (e.g. Kaisse, 2002; Mielke, 2008). In line with this observation, I use the term *phonological class* throughout this paper, rather than *natural class*, to refer to a set of sounds that behave in a uniform way in the phonology of a language without necessarily sharing any phonetic properties.

These observations have led some researchers to propose that phonological classes may be learned and language-specific (e.g. Mielke, 2008; MacWhinney & O'Grady, 2015; Archangeli & Pulleyblank, 2015). Under this view, phonologically salient classes need not be phonetically coherent, and distributional learning accounts for a much larger part of phonological acquisition than previously thought. The observation that classes tend to be phonetically coherent cross-linguistically can be explained by a tendency for similar sounds to undergo similar diachronic processes that lead to this patterning (e.g. Blevins, 2004).

It may be the case that these phonetically disparate classes can be captured by other means. Interactions between phonological processes that target phonetically coherent classes may result in what superficially appear to be unusual classes. Alternatively, these classes may be coherent with respect to a phonetic dimension that has not yet been included in current feature systems. Regardless of whether one is willing to commit to the position of emergent classes, these ideas raise theoretically interesting questions. Namely, to what extent are phonological classes apparent in the distribution of sounds in a language, and to what extent might learners use this information?

This paper will investigate the learning of phonological classes when *only* distributional information is available. It will do so by detailing an algorithm that attempts to learn as much phonological information as possible solely from the contexts in which sounds do and do not occur. This is not to suggest that phonetic information does not play an important role in characterizing phonological classes: rather it is an attempt to see how far we can get when restricting ourselves to only one of the many sources of information available to the learner.

From a high level, the algorithm consists of two components. In the first stage, sounds in a phonological corpus are projected into a vector space based on their distributional properties. In the second stage, a clustering technique is used on these vector representations to retrieve a set of classes. This is done by performing Principal Component Analysis and then applying one-dimensional k-means clustering to the resulting principal components to retrieve classes. The procedure is then recursively performed on the discovered classes.

Aside from eschewing phonetic information, this algorithm operates under two additional assumptions. First, it focuses only on phonotactic information: there is no explicit attempt to capture alternations. Although this may be a reasonable assumption about the initial phonological learning done by infants, it is expressly adopted here as a simplifying assumption. More sophisticated models may benefit from incorporating this information. Second,

because it takes phonemic text corpora as input, it necessarily assumes that the learner has access to a segmental representation of speech.

The output of the algorithm is a set of phonological *classes* that may be viewed as implicitly reflecting a feature system, in that any class contained in this set can be uniquely characterized by some combination of feature/value pairs. The process of deriving an explicit feature system from a set of classes is described in a related paper (Mayer & Daland, submitted).

The paper is structured as follows. Section 2 reviews past research that has taken a distributional approach to learning linguistic structure. Section 3 describes a toy language with well defined phonotactic properties, which will serve as a running example throughout the paper and a basic test case for the algorithm. The next two sections describe the components of the algorithm. Section 4 details how a vector space representation of the sounds of a language can be generated from a phonological corpus. Section 5 shows how a combination of Principal Component Analysis and clustering algorithms can be used to extract potential phonological classes from such embeddings, and details its performance on the toy language. Section 6 presents the results of its application to Samoan, English, French, and Finnish. It is able to successfully distinguish consonants and vowels in every case, and retrieves interpretable classes within those categories for each language. Finally, Section 7 compares these results against past work, and Section 8 offers discussion of the results and proposals for future research.

2 Previous work

Distributional learning has been proposed as a learning mechanism in most areas of linguistics, suggesting that it may be a domain-general process. Examples include word segmentation and morphology (e.g. Saffran et al., 1996; Goldwater et al., 2009; Goldsmith, 2010), syntax (e.g. Redington et al., 1998; Wonnacott et al., 2008), and semantics (e.g. Andrews et al., 2009; Bruni et al., 2014).

Distributional approaches to phonology have been explored since the early part of the 20th century, but as Goldsmith and Xanthos (2009) point out, most of this work is not well known today. A number of pre-generative phonologists discussed the merits of such approaches and potential implementations, but this work was necessarily limited by technological factors.[1] The increasing availability of ever more powerful computers together with advances in statistical and machine learning research have recently rendered such approaches more viable.

Powers (1997) provides an extremely detailed empirical comparison of early work building on these advances. Notable additions include the abstraction of representing sounds as points in a high dimensional space (see Section 4), and the idea of using matrix factorization and bottom-up clustering algorithms to group sounds together. While these approaches were a notable step forward, they frequently failed to achieve the basic distinction between the consonants and vowels of a language. This should be taken with some caution, however, as Powers ran his evaluations on orthographic rather than phonemic data.

[1] See Appendix A in Goldsmith and Xanthos (2008) for a detailed summary of this work.

In the same time period, Ellison (1991, 1994) explored a *minimum description length* analysis, which uses an information theoretic objective function to evaluate the success with which a set of classes fits an observed data set. The optimal candidate set of classes is found using simulated annealing. Ellison reports that his method is generally successful in differentiating consonants and vowels across a wide range of languages, as well as identifying aspects of harmony systems. Ellison also runs his models on orthographic data.

More recently, Goldsmith and Xanthos (2009) compared three different approaches to learning phonological classes in English, French, and Finnish. The first, Sukhotin's algorithm, is mostly of historical interest, but can differentiate between consonants and vowels reasonably well using calculations simple enough to be performed by hand. Their second approach uses *spectral clustering*, which models distributional similarity between segments as an undirected graph with weighted edges. By representing the graph as a matrix and using eigendecomposition to find an optimal partition into two or more groups, Goldsmith & Xanthos were able to successfully achieve a distinction between consonants and vowels, and a basic characterization of harmony systems. The final approach they examine is *maximum likelihood hidden Markov models*. These use a finite state machine with some small number of states (e.g. two for vowel vs. consonant). The model is trained to calculate transition and emission probabilities that maximise the likelihood of the data. The ratio of emission probabilities for each segment between states can then be used to classify them. This approach worked well for distinguishing vowels and consonants, identifying vowel harmony, and (to some extent) syllable structure.

Calderone (2009) used a similar approach to spectral clustering, *independent component analysis*, which tries to decompose a matrix of observed data into a mixture of statistically independent, non-Gaussian components. This resulted in a qualitative separation between consonants and vowels, as well as suggesting some finer grained distinctions within these sets.

Taking a different approach, Nazarov (2016) details an algorithm for jointly learning phonological classes and constraints using a combination of maximum entropy learning and Gaussian mixture models. Segments that are targeted by constraints that refer to a similar context are hypothesised to form a natural class, and specific constraints are in turn combined into more general ones using these classes. This performs reasonably well on a simple artificial language.

Peperkamp et al. (2006) also use distributional patterns to learn phonological information, although here they attempt to find allophonic relationships rather than phonological classes. They use the Kullback-Leibler divergence to identify pairs of sounds that rarely occur in the same contexts (i.e. in complementary distribution). This is able to find true allophones successfully, but also introduces many spurious allophones, and they must make use of a phonetic similarity filter to rule out these cases.

Finally, models that learn phonetic and phonological classes from acoustic (e.g. Lin, 2005) and articulatory (e.g. Mielke, 2012) data have been proposed. Although the input data differs, the techniques for finding classes used in these studies are similar to the ones employed here. A more complete model of the learnability of phonological classes should refer to both types of data.

The goal of this paper is to expand on the successes of previous work that has attempted to learn phonological classes from distributional information alone (e.g. Goldsmith & Xan-

thos, 2009; Calderone, 2009; Nazarov, 2016). These methods have generally been effective in finding a reliable distinction between consonants and vowels, and some simple partitions of these sets (e.g. front vs back vowels in Finnish, which has vowel harmony along this dimension). Nazarov (2016) is able to learn more complex classes from a very simple toy language, but it is unclear to what extent this generalises to natural languages. I will show that the algorithm presented here successfully finds classes that stand in a complex relationship to one another in an artificial language, is more successful in learning finer-grained categories in natural languages than these past approaches, and provides a deterministic, rather than qualitative, method for identifying classes. The basic structure of first performing vector embedding of the sounds in a corpus followed by clustering to retrieve classes also provides a useful general framework in which further studies of distributional learning might proceed. Finally, the code implementing this algorithm is publicly available, and researchers are encouraged to use and modify it for their own purposes.[2]

3 Parupa: An artificial language

Because it is not clear a priori what classes might be apparent in the distribution of a natural language, it is useful to begin with a case where the target classes are known in advance. To this end, I introduce an artificial language called *Parupa*, which has well-defined phonotactic properties. Parupa serves as a running example throughout the paper and an initial test case for the algorithm. Its consonant and vowel inventories are shown in Tables 1 and 2.

p	t	k
b	d	g
	r	

Table 1: Parupa consonants

i		u
e		o
	a	

Table 2: Parupa vowels

Parupa has the following distributional properties:

1. All syllables are CV.
2. Vowel harmony: words must contain only front (/i/, /e/) or back (/u/, /o/) vowels. /a/ may occur in either case (i.e. it is transparent to harmony).
3. Words must begin with /p/ or /b/.

[2]The source code can be found at https://github.com/connormayer/distributional_learning.

4. Consonant-vowel co-occurrence restrictions: /p/, /t/, and /k/ must be followed by high vowels or /a/. /b/, /d/, and /g/ must be followed by mid vowels or /a/. /r/ may be followed by any vowel. In other words, the full set of consonants is only in contrast before /a/.

These particular properties were chosen to include multiple, overlapping partitions of the sets of vowels and consonants. For example, the vowel set is partitioned in two different ways: high-mid, and front-back. This structure is common in natural languages, and introduces challenges for many clustering algorithms (see Section 5). Given these properties, the algorithm should retrieve the classes shown in Figure 1.

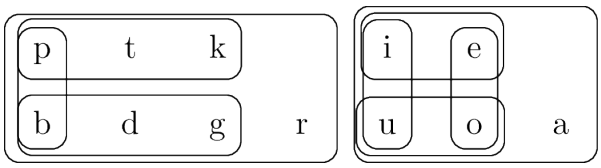

Figure 1: The phonological classes of Parupa.

A language corpus was generated using a Hidden Markov Model, shown in Figure 2. Although all emission and transition probabilities for any state were equal, the phonotactic constraints meant that not all segments were equally common in the corpus (e.g. /a/ was the most frequent vowel). The generated corpus had 50k word tokens, which resulted in a total of about 18k word types. The input to the algorithm that will be described consists only of the word types.[3] The average word length was three syllables. Examples of Parupa words are shown in Table 3.

berari
pupabopa
pa
paka
boka
padoropa
bo
pakubatuda
bopu
piretiba
pabarubo
barika

Table 3: Some Parupa words

I will use Parupa as a running example throughout the rest of the paper to illustrate the performance of the various components of the algorithm.

[3]Consistent with previous phonological modeling done over corpora like Bybee (1995), type frequency produces more interpretable results than token frequency, and is used throughout. In other words, the corpora employed are dictionary-like lists of word types rather than texts containing multiple word tokens.

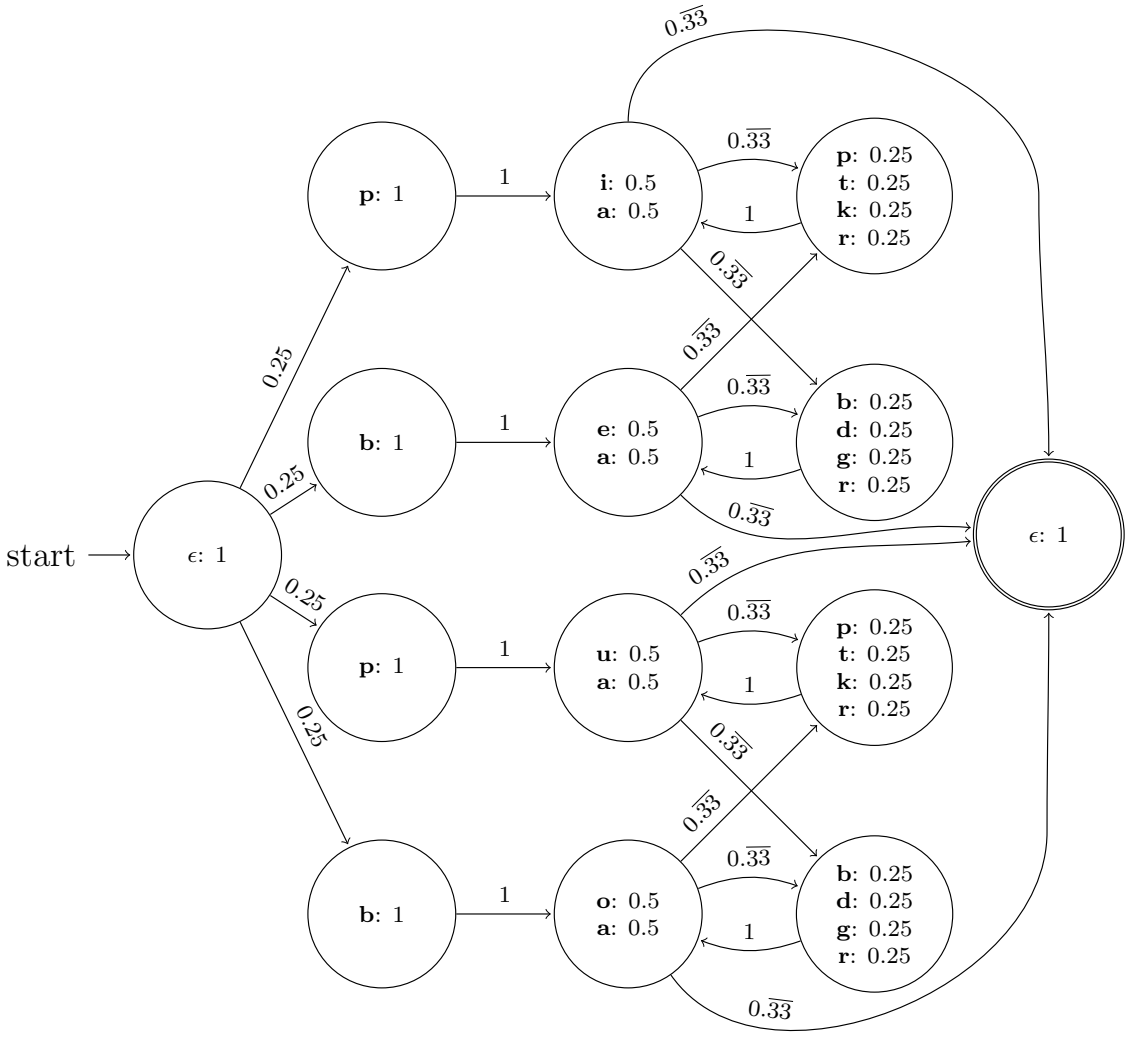

Figure 2: The Hidden Markov Model used to generate Parupa. Edges are labeled with their transition probabilities, and states are labeled with their segment emission probabilities. ϵ is the empty string.

4 Quantifying similarity: Vector space models

This model operates under the assumption that similar sounds[4] in a language should have similar *distributions*. A distribution is a description of how frequently each outcome in a set of possible outcomes is observed in a data set. In this case, the outcomes we are interested in are the *contexts* in which a sound occurs: i.e. the other sounds that occur near it.

To generate distributions for each sound, I adopt *vector space modelling* (e.g. Manning & Schütze, 1999; Jurafsky & Martin, 2008). The principle behind this approach is to represent objects as vectors or points in an n-dimensional space whose dimensions reflect some of their properties. Embedding objects in a vector space allows for convenient numerical manipulation and comparison between them.

This approach is commonly applied in many language-related domains: in document retrieval, where documents are represented by vectors whose dimensions reflect words that occur in the document; in computational semantics, where words are represented by vectors whose dimensions reflect other words that occur near the target word; and in speech/speaker recognition, where sounds are represented by vectors whose components are certain acoustic parameters of the sound. This is also essentially the approach taken by many of the papers discussed in the previous section, where sounds are represented as vectors whose dimensions reflect the counts of sounds that occur near them. Whether we are dealing with documents, words, or sounds, the projection of these objects into a vector space should be done in such a way that similar objects end up closer in the space than less similar ones.

An important distinction between applying this approach to documents or words and applying it to sounds is that order is crucially important for sounds. When considering the semantics of words or documents, it is generally more useful to know that a word occurs in a document or that a word occurs near another word than it is to know that a word is the nth word in a document, or that a word occurs exactly n words before another word. In contrast, ordering is crucial for phonology, since adjacency and directionality play important roles in phonological processes.[5]

The methods I use here combine aspects of the approaches described above. Before going into more detail, I will first provide a simple, concrete example of how we can go from a phonological corpus to a vector representation of each sound in the language.

4.1 A simple vector embedding of sounds

Suppose we have a language with only two sounds, /t/ and /a/, and a corpus containing the following five words:

ta, ata, tata, atta, taa (1)

Σ is the set of all unique symbols in the corpus, plus the special symbol #, which represents a word boundary.[6] Here $\Sigma = \{t, a, \#\}$.

[4] I intentionally use the word "sounds" rather than "phonemes" or "phones" because this model is independent of the level of transcription used in the corpus.

[5] Not all aspects of ordering are important for phonology: knowing that a sound is the third sound in a word is not generally useful, although knowing that a sound is first or last in a word can be.

[6] For clarity, I omit word boundaries in the presentation of the data here.

To go from this corpus to a vector space representation of each of the sounds, we must decide how we want to define the dimensions of the resulting vector space: i.e. which aspects of context we wish to be sensitive to, and how to quantify these aspects. For this simple example, I will define each dimension in the space as the number of times a particular symbol occurs immediately *before* the target symbol. That is, the corresponding vector for each symbol in Σ (except for #) consists of dimensions with labels s_i_, where _ indicates the position of the target sound (the sound whose vector we are constructing), s_i ranges over Σ, and the value of each element is the number of times s_i occurs before the target sound in the corpus.[7] In general when discussing vectors, I will use _ to indicate the position of the target sound, and s with subscripts to indicate sounds in the context.

The resulting count vectors under this definition are shown in Table 4.

	t_	a_	#_
t	1	3	3
a	6	1	2

Table 4: Count vectors for a toy language.

For example, the cell in the bottom left corner of this table has the value 6 because /a/ occurs after /t/ six times in the corpus. Note that although these sounds have overlapping distributions, these vectors capture the general pattern of alternation between the two. It is straightforward to see how these counts can be interpreted as points or vectors in 3D space, where $t = (1, 3, 3)$ and $a = (6, 1, 2)$.

4.2 What do we count when we count sounds?

The previous example counts the sounds that occur immediately preceding the target sound. This is not likely to be informative enough for identifying phonological classes in anything but the simplest languages. There are many other ways we might choose to count contexts. Here I adopt *trigram* counting, which counts all contiguous triples of sounds that contain the target sound. Thus our dimension labels will be of the form $s_i s_j$_, s_i_s_j, and _$s_i s_j$, where s_i and s_j range over Σ. Under this counting scheme, the number of dimensions is $3|\Sigma|^2$, where Σ includes the word boundary symbol. A discussion of the limitations of this counting scheme and some other possibilities will be presented in Section 8.

4.3 Weighting counts

Raw counts tend to not be particularly useful when dealing with vector embeddings of words, because many different types of words can occur in the same contexts (e.g. near *the* or *is*). A common technique is to somehow weight the counts, such as by converting them to probabilities, conditional probabilities, or more sophisticated measures. Weighting proves to be valuable for sounds as well. The basic assumption I make is that the most

[7] These dimension labels should be considered analogous to the convention of using the labels x, y, and z to refer to the axes in 3-dimensional space.

fundamental partition of the sounds in any language should be between consonants and vowels (or alternatively, sounds that occupy syllable nuclei and sounds that do not). A suitable weighting method should make this distinction apparent. Of the weightings tested, only *Positive Pointwise Mutual Information* (PPMI) was able to consistently produce a clean distinction between consonants and vowels across data sets.

PPMI is an information theoretic measure that reflects how frequently a sound occurs in a context compared to what we would expect if sound and context were independent (Church & Hanks, 1990). It is defined as follows

$$PPMI(s,c) = max(log_2 \frac{Pr(s,c)}{Pr(s)Pr(c)}, 0) \tag{2}$$

where s is a sound and c is a context. The calculations for the three probabilities in the equation are given below, where a *token* is a particular occurrence of a sound.

$$P(s) = \frac{\text{\# of tokens of } s}{\text{total \# of tokens}} \tag{3}$$

$$P(c) = \frac{\text{\# of tokens in context } c}{\text{total \# of tokens}} \tag{4}$$

$$P(s,c) = \frac{\text{\# of tokens of } s \text{ in context } c}{\text{total \# of tokens}} \tag{5}$$

These values are easily computable from a corpus.

If $Pr(s)$ and $Pr(c)$ are independent, then $Pr(s,c) \approx Pr(s)Pr(c)$ and hence the value of the inner term $log_2 \frac{Pr(s,c)}{Pr(s)Pr(c)}$ will be close to 0. If $P(s,c)$ occurs more frequently than the individual probabilities of s and c would predict then the value will be positive, and if $P(s,c)$ occurs less frequently than expected, then this term will be negative.

PPMI converts all negative values of the inner term to 0 (as opposed to *Pointwise Mutual Information*, which does not (Fano, 1961)). This is desirable when dealing with words rather than sounds, because the size of the vocabulary often requires an unreasonable amount of data to distinguish between words that tend not to co-occur for principled reasons and words that happen not to co-occur in the corpus (e.g. Dagan et al., 1993; Niwa & Nitta, 1994). Although this seems as though it should be less of a concern with phonological data given the relatively small number of sounds, in practice PPMI appears to provide more interpretable results than PMI on the data sets examined here.[8]

Table 5 shows the count vectors from the toy corpus in the previous section converted to PPMI. The vectors have been smoothed, in a sense, with the separation between the two on each dimension becoming even more pronounced.

[8]This result appears to be at odds with the centrality of markedness constraints in phonological theory. I suspect that, as for words, the number of coincidentally unattested sequences of sounds overwhelms the number of sequences that are prohibited by markedness constraints or the like. For example, the English CMU pronouncing dictionary is transcribed using 39 phonemes, and contains 27,209 words of length six. There are $39^6 = 3,518,743,761$ possible words of length six that could be generated from an inventory of 39 phonemes. This means that attested six sound words only make up about 0.0007% of possible six sound words. Because there are so many unattested sequences, it may be the case that it is more informative to know where sounds do occur than where they do not. I leave a detailed exploration of this as a topic for future research.

	t_	a_	#_
t	0	0.78	0.46
a	0.61	0	0

Table 5: Count vectors for a toy language weighted using PPMI.

4.4 PPMI Vector Embeddings of Parupa

A challenge in dealing with high dimensional spaces is visualizing the data. Here I use Principal Component Analysis (PCA) (Hotelling, 1933), which projects points in a space onto a smaller set of dimensions, called principal components (PC), such that the variance of the projected data is maximised. This technique is useful for reducing high dimensional spaces to two or three dimensions so they can be visualised. It will also be of crucial importance in the clustering stage described in Section 5, and a more detailed description will be given there.

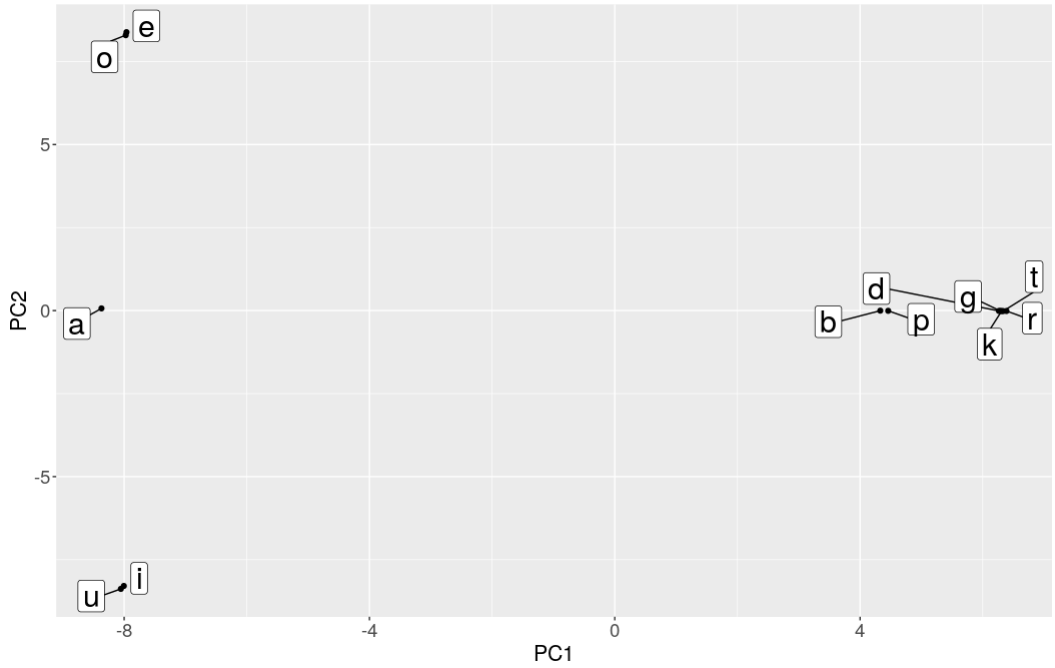

Figure 3: A PCA visualization of the vector embedding of Parupa consonants and vowels using trigram counts and PPMI weighting.

Figure 3 shows a two-dimensional PCA visualization of the vector space embedding of Parupa using trigram counts and PPMI weighting. Here we can see that the vowel/consonant distinction is clear along PC1, and vowel height is reflected on PC2.[9]

Figures 4 and 5 show PCAs generated with only consonants and only vowels respectively. For the consonants, the distinction between sounds that must precede high vowels and sounds

[9]The reader should keep in mind that referring to a phonetic property here is a shorthand for referring to particular aspect of the distribution, since there is no notion of phonetic substance in this model.

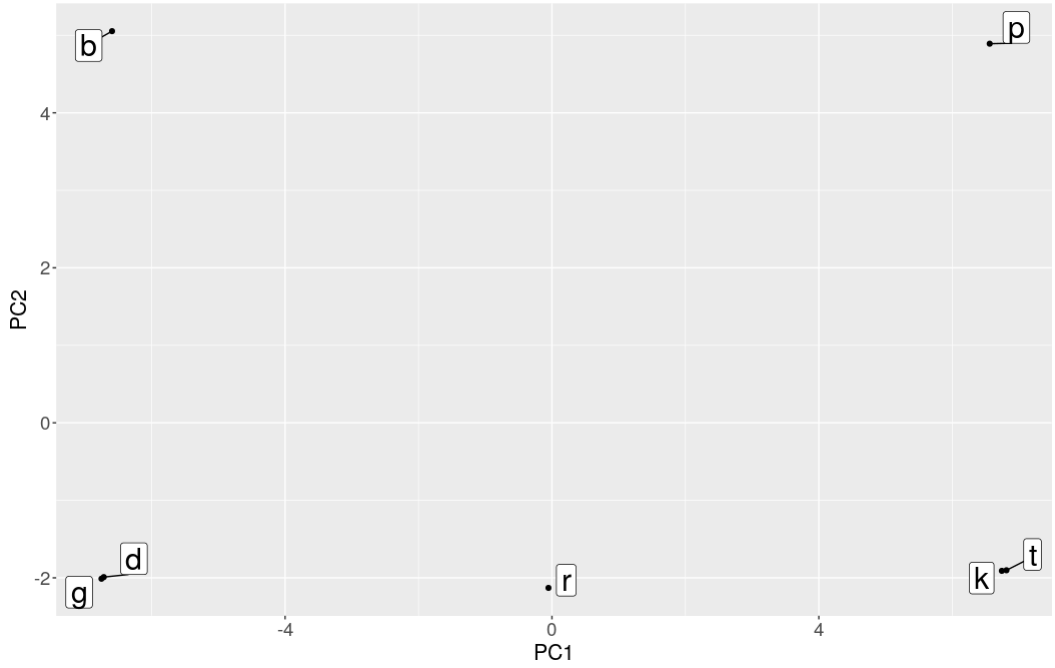

Figure 4: A PCA visualization of the vector embedding of Parupa consonants using trigram counts and PPMI weighting.

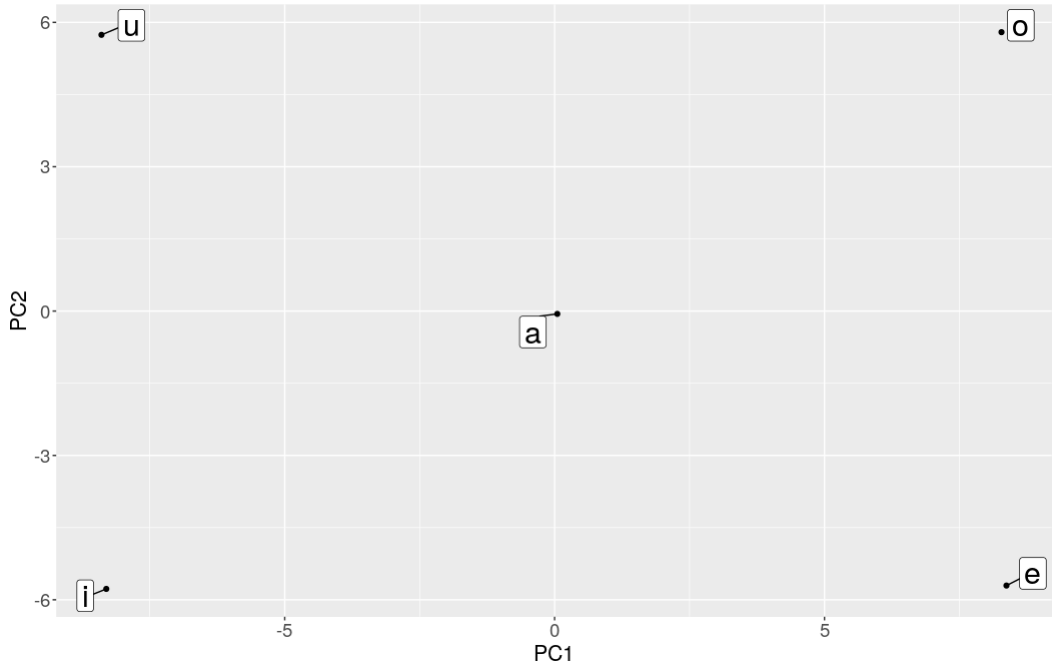

Figure 5: A PCA visualization of the vector embedding of Parupa vowels using trigram counts and PPMI weighting.

that must precede mid vowels is reflected in PC1, while the distinction between sounds that can begin a word and sounds that cannot is reflected in PC2. For the vowels, the height distinction is reflected on PC1, while the backness distinction is reflected on PC2. Note the intermediate position of /r/ and /a/ in the plots, reflecting their shared distributions within the consonant and vowel classes.

PCA visualizations must be interpreted with caution, since they generally lose information present in the full space. In the simple case of Parupa, however, it seems clear that there should be sufficient information in the vector embedding to retrieve the intended classes.

5 Finding classes using PCA and k-means clustering

Once we have vector space embeddings of the sounds in our corpus, we need a way to extract phonological classes from the space. It is intractable to consider every possible set of classes, since given an alphabet Σ, there are $2^{|\Sigma|}$ possible classes, and hence $2^{2^{|\Sigma|}}$ sets of classes that could be chosen. One approach to generating a reasonable set of candidate classes is using *clustering algorithms*. Broadly speaking, such algorithms attempt to assign each point in a data set to one or more clusters, such that the points in each cluster have more in common with other points in the cluster by some criterion than with points outside of the cluster.

Many clustering algorithms with different properties and assumptions have been proposed,[10] but the nature of the problem of finding phonological classes imposes several restrictions on the type of algorithm that should be used.

1. It must be *unsupervised*, meaning that the algorithm requires no access to training data (i.e. sounds that have already been assigned to classes).

2. It must not require the number of classes to be specified in advance.

3. It must allow *multiple class membership*. This is analogous to saying that it must allow a set of sounds to be partitioned in multiple ways. In Parupa, for example, /i/ patterns as both a front vowel and a high vowel.

4. Distributional evidence for class membership may be present only in some contexts. For example, the high/mid vowel distinction in Parupa is signaled only by the preceding consonant, while the front/back distinction is apparent only from the preceding and following vowels. A suitable clustering algorithm should be able to look at meaningful subsets of all contexts when clustering sounds.

There are clustering algorithms that meet these criteria, particularly certain *subspace clustering* algorithms (e.g. Müller et al., 2009), but properties of the data considered here make them difficult to apply for practical reasons. First, these algorithms are generally difficult to parameterise in a principled way, requiring assumptions about the number of clusters or the distributional properties of the data. Second, phonological data by definition has no outliers (i.e. all sounds should be clustered), but many common clustering algorithms assume their presence. Finally, our data consist of a small number of points, one per sound,

[10] See e.g. Aggarwal and Reddy (2013) for an overview.

and a large number of dimensions, one per context. Most clustering algorithms are optimised to handle the opposite situation well, and this leads to severe efficiency issues.

In light of these problems, I propose a clustering technique that is well suited to this task. It works by recursively applying Principal Component Analysis and one-dimensional k-means clustering. The next sections will show that this combination allows for multiple partitions of the same set of data, while simultaneously exploiting the generally hierarchical structure of phonological classes.

5.1 Principal Component Analysis

Principal Component Analysis is a dimensionality reduction technique. I will not describe its formal properties here, but treatments of it can be found in almost any introductory statistics textbook (I use Everitt & Dunn, 2001). It takes a matrix consisting of some number of (possibly correlated) dimensions and reduces it to a set of new, uncorrelated dimensions called principal components. These principal components are linear combinations of the original dimensions. The number of principal components is $min(m-1, p)$, where m is the number of rows in the data set and p is the original number of dimensions. Principal components are ordered descending by proportion of variance captured, with PC1 capturing the most variance, followed by PC2, and so on. This has several useful consequences:

1. To reduce a data set to n dimensions while minimizing the amount of information lost, we can simply choose the first n principal componentss.

2. Because we know how much variance each principal component captures, we can choose the number of new dimensions using a variance-based criterion. This could be choosing the number of dimensions required to capture some percentage of the original variance, or choosing only dimensions whose variance contribution exceeds some threshold.

We can run PCA on a matrix whose rows are the vector embeddings of individual sounds in a corpus. PCA is useful for clustering phonological data for several reasons: first, because our matrix consists of few rows and many dimensions, its dimensions are highly correlated. Applying PCA reduces the matrix to a set of uncorrelated dimensions, which makes interpretation more straightforward. Second, PCA helps to highlight robust sources of variance while reducing noise. Finally, the resulting principal components provide some insight into the different ways to partition a set of sounds. Consider again Figure 5. PC1, which captures the largest proportion of the original variance, shows the distinction between high and mid vowels while revealing little about the front/back vowel distinction. This distinction is apparent in PC2, however. Thus looking at different principal components has the potential to expose multiple ways to partition a single set of sounds.

5.2 k-means clustering

Given a principal component, we would like to determine how many classes the distribution of sounds suggests. In Figure 4, for example, a visual inspection suggests PC1 should be grouped into three classes: {b,d,g}, {r}, and {p,t,k}, while PC2 should be grouped into two classes: {b,p} and {d,g,r,k,t}. k-means clustering can be used to optimally group a set of

points into k clusters by finding cluster centers and assigning points to clusters in such a way that the total distance from each point to its cluster center is minimised (MacQueen, 1967). In order to determine the optimal value of k, information theoretic measures such as the Akaike Information Criterion (AIC) (Akaike, 1974) or Bayesian Information Criterion (BIC) (Schwarz, 1978) can be used. These measures attempt to strike a balance between model complexity and model fit by penalizing more complex models (in this case, higher values of k) while rewarding fit to the data (in this case, distances from the cluster centers).

Because we are clustering on principal components which are one-dimensional, I use the *Ckmeans* algorithm (Wang & Song, 2011), which is an optimization of the standard k-means algorithm for one-dimensional data. This algorithm efficiently finds the optimal number of clusters using the BIC as an evaluation metric. When applied to PC1 and PC2 of the set of consonants discussed in the previous paragraph, this algorithm finds exactly the expected classes: namely {b,d,g}, {r}, and {p,t,k} on PC1, and {b,p} and {d,g,r,k,t} on PC2.

Readers familiar with clustering techniques might find it odd that clustering is done over single principal components rather than all dimensions, whether these be the original dimensions representing specific contexts, or the reduced dimensions after PCA is performed. This is a sensible choice because of the properties of vector embeddings of sounds in a phonological corpus and the nature of phonological classes in general.

First, as mentioned earlier, the columns in the vector space are massively redundant. Each principal component in a PCA can be thought of as an aggregation of the information in a correlated set of columns in the original data. Put another way, PCA does some of the work of finding meaningful subspaces of the vector space over which clustering is likely to be effective, and thus each principal component can be thought of as representing some number of dimensions in the original space.

Additionally, clustering over individual principal components rather than sets of principal components allows us to find broad classes in the space that might otherwise be overlooked. This is apparent when examining Figure 5: clustering over PC1 and PC2 separately allows us to find distinct partitions of the vowel space based on height and on backness. If PC1 and PC2 were considered together, the only likely clusterings would be either a single cluster containing all vowels, missing the class structure completely, or one cluster per sound. The latter is equivalent to finding classes that reflect the *intersections* of different height and backness values, but overlooks the broader class structure from which these subclasses are generated. Finding such classes is a property that many subspace clustering algorithms have, but, as described above, many of these algorithms are unsuited to this type of data for a variety of reasons. Clustering over single principal components is simple way to achieve this property while circumventing many of these issues.

5.3 Recursively traversing the set of classes

The final component of this clustering algorithm leverages the generally hierarchical nature of phonological classes. In many cases a distinction is only relevant to segments in a particular class: for example, the feature $\left[\text{ +/− strident } \right]$ is only relevant for coronal fricatives and affricates. Expressed in a slightly different way, patterns that do not contribute a great deal to the variance of the entire set of sounds might become more apparent when only a subset of the sounds is considered. In order to exploit this hierarchical structure, this clustering

algorithm is called recursively on the sets of classes that are discovered on each principal component.

5.4 Putting it all together

To summarise, this algorithm runs Principal Component Analysis on a matrix of vector embeddings of sounds and attempts to find clusters on the most informative principal components. For each cluster found, the algorithm is recursively applied to that cluster to find additional subclusters. Considering multiple principal components for each set of sounds has the potential to partition every set of sounds in multiple ways, and the recursive character allows it to exploit the generally hierarchical nature of phonological classes to discover more subtle class distinctions.

The steps of the algorithm and the necessary parameters are detailed below:

1. Initially use the original vector embedding matrix as input data.
2. Perform Principal Component Analysis on the input data.
3. For each principal component, PCi, where $1 \leq i \leq n$:
   (a) Cluster the sounds on PCi into between 1 and k clusters.
   (b) If more than one cluster is found, run this algorithm again on each cluster (i.e. return to step 2), using as input data only the rows of the original vector embedding matrix corresponding to the sounds found in the cluster.
4. Return the clusters that were found by this and all recursive calls.

The two parameters that must be set here are n, the number of principal components we consider for each input, and k, the maximum number of clusters we attempt to partition each principal component into.

k is relatively easy to choose if we assume the typical properties of phonological feature systems, where a class is either +, −, or 0 (unspecified) for a particular feature. This suggests we should partition each principal component into either one (no distinction), two (a +/− or +/0 distinction, as in PC2 in Figure 4), or three (a +/−/0 distinction, as in PC1 in Figure 4). Thus, setting $k = 3$ seems like a principled choice.

When choosing n, we want to select only those principal components that are informative about meaningful phonological classes. If n is too high, principal components that contain mostly noise will be included and result in spurious classes being detected. If n is too low, important classes may be overlooked. There have been many proposals for how to choose the number of components to consider (see e.g. Section 3.5 in Everitt & Dunn, 2001). Here I use the relatively simple Kaiser stopping criterion (Kaiser, 1958), which suggests taking only the principal components that account for above-average variance (i.e. whose eigenvalues are greater than the average of the eigenvalues of all principal components). This criterion is simple to calculate and works well in practice here. In general, however, the choice of which components to use should be thought of as a parameter that might be tuned for different purposes (e.g. we might want to consider less robustly-attested classes with the intention

of later evaluating them on phonetic grounds). Increasing or decreasing the number of components used has the effect of increasing or decreasing the algorithm's sensitivity to noise, and determines how robust a pattern must be to be retrieved.[11]

In the next section, I present the results on Parupa to illustrate the algorithm's effectiveness.

5.5 Running the algorithm on Parupa

Running the algorithm on the Parupa vector embeddings detailed in Section 4 produces the classes in Table 6:

{a, i, u, e, o}	**{p, t, k, b, d, g, r}**
{i, u}	**{b, d, g}**
{e, o}	**{p, t, k}**
{i, e}	**{p, b}**
{u, o}	{d, g, k, r, t}
{a}	{d, g}
	{k, t}
	{p}
	{b}
	{r}

Table 6: Classes learned from Parupa. Bolded classes indicate predicted classes.

All the expected classes are present in this set, and although there are additional classes, these are derivable from the expected classes: e.g. {d,g,k,r,t} is the class of non-word-initial consonants, {d,g} is the class of non-word-initial consonants that can precede mid vowels, {k,t} is the class of non-word-initial consonants that can precede high vowels, etc. The hierarchical relationship between these classes is shown in Figure 6, which was generated using code from Mayer and Daland (submitted). These diagrams are used throughout the paper, and do not reflect the order in which the classes were retrieved by the algorithm: rather, they arrange the classes in a hierarchical structure, where arrows between classes represent a parent-child relationship (i.e. the child class is a proper subset of the parent class, and there is no other class that intervenes between the two). Dotted arrows indicate that a class is the intersection of two or more parents. In essence, this diagram gives a sense of the overall relationship between the classes retrieved by the algorithm, rather than the path the algorithm took to retrieve the classes.

Note that the singleton classes consisting of individual segments are not in general retrieved by the algorithm. This is the consequence of the k-means clustering component deciding that no partition of a class into two or three classes is warranted. This is not of

[11] It may be the case that this parameter can be chosen based on phonological criteria by looking at how many different partitions of a single set of sounds are typical in natural languages. I leave this as a topic for future research.

great concern, however, since the assumption of a segmental representation necessarily implies that singleton classes are available to the learner. These may simply be appended to the list of retrieved classes if so desired.

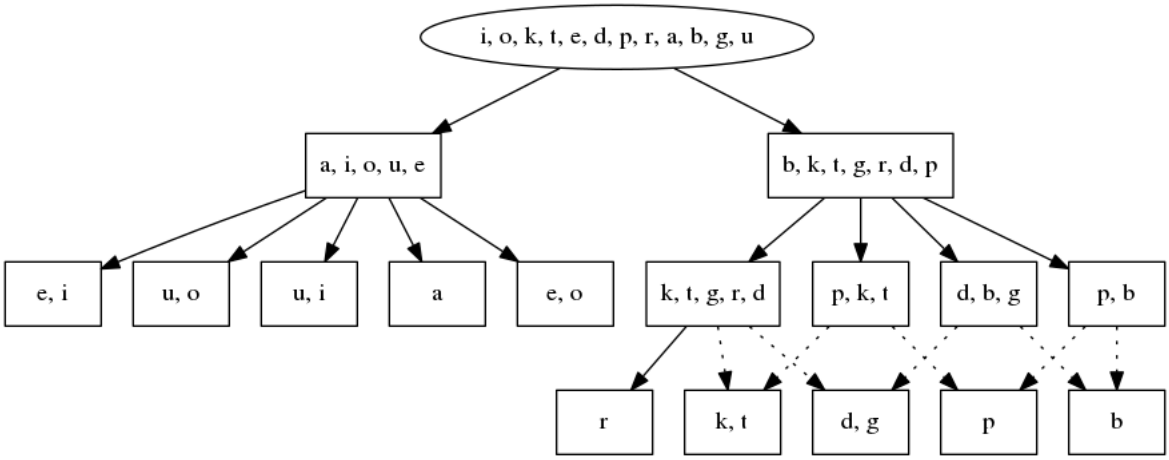

Figure 6: The classes retrieved from Parupa. Arrows indicate a parent/child relationship between classes. Dotted arrows indicate that a class is the intersection of two or more parents.

This algorithm performs well on Parupa, successfully retrieving all of the intended classes, including those that involve partitioning sets of sounds in multiple ways.

5.6 Evaluating the robustness of the algorithm on Noisy Parupa

Parupa is a pathologically tidy language: its phonotactic constraints are never violated. Although the algorithm does well on retrieving the class structure to which these constraints are sensitive, no natural language is so well behaved. In order to evaluate how well the algorithm described above handles noise, I examine its performance on a more unruly variant of Parupa: Noisy Parupa.

Noisy Parupa is identical to Parupa, except that some percentage of the generated word tokens are *noisy*: they do not conform to the phonological restrictions outlined in Section 3. These words still maintain a CV structure, but the consonants and vowels in each position are chosen with uniform probability from the full sets of consonants and vowels. A Hidden Markov Model for generating noisy words is shown in Figure 7. Transition probabilities were chosen so that the average word length is still three syllables.

A parameter determines what percentage of the words are noisy. Standard Parupa can be thought of as a special case where this parameter is set to 0. As the value of this parameter increases, the algorithm should have more difficulty finding the expected phonological classes.

The model was tested on 110 corpora. The noise parameter was varied from 0% to 100% in increments of 10%, and ten corpora were generated for each parameter value.

Figure 8 shows the median number of expected and unexpected classes found by the algorithm as the percentage of noisy words increases.[12] The expected classes are defined

[12]The mean number of classes was similar.

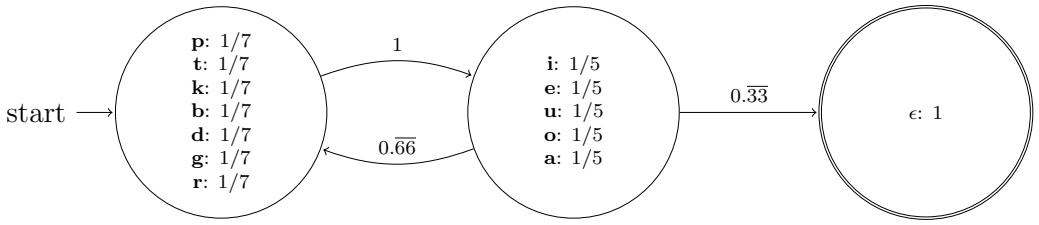

Figure 7: The Hidden Markov Model used to generate Noisy Parupa words. Edges are labeled with their transition probabilities, and states are labeled with their segment emission probabilities. ϵ is the empty string.

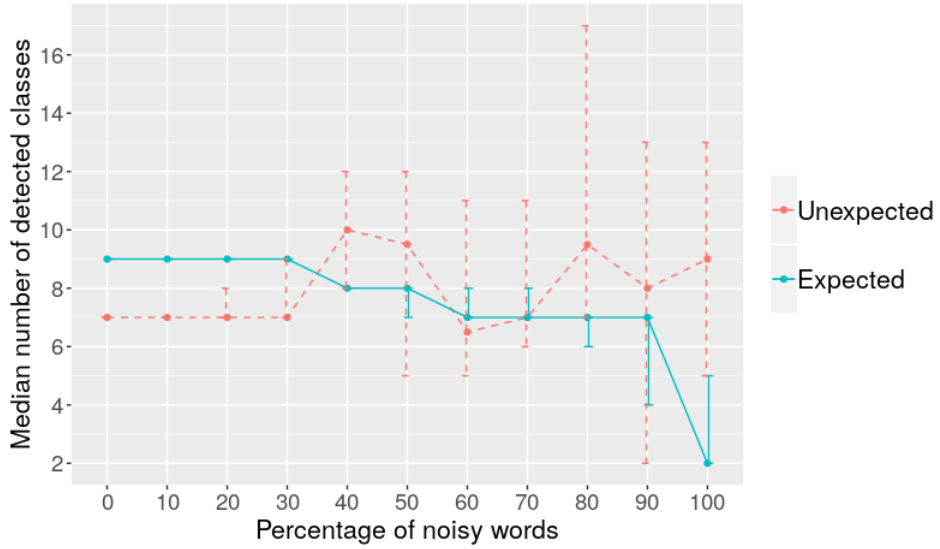

Figure 8: A plot of the median number of expected and unexpected classes found by the algorithm as the percentage of noisy words increases. Error bars span the minimum and maximum number of classes retrieved from a corpus at that noise level.

as exactly the classes in Figure 1. The number of unexpected classes varies as the specific properties of the corpus change, but the number of expected classes found remains reasonably high until 100% noise. From 40% to 70% noise, the expected classes that are not detected are either {k,p,t}, {b,d,g}, or both. In about half the cases (19/40) the unexpected classes include {k,p,t,r} and/or {b,d,g,r}. In 20 of the remaining 21 cases, the sets {p,t} and/or {b,g} are recovered. This indicates that the pattern is still detected to some extent, although the participating classes are less clear due to the increase in noise.

From 80% to 90% noise, the algorithm reliably fails to detect the classes {k,p,t} and {b,d,g}, while occasionally also overlooking other classes: {b,p} (4/20), {o,u} (3/20), {i,u} (1/20), {e,i} (3/20) and {e,o} (1/20).

Finally, at 100% noise, the consonants and vowels are the only classes reflected in the distribution, and these are successfully retrieved in all cases. The other expected classes that are sometimes retrieved are the result of chance.

The results of the algorithm on Noisy Parupa suggest that it is fairly robust to noise. All expected classes are discovered in up to 30% noise, and even up to 90% noise most of the

expected classes are still found. Even when expected classes are lost at higher noise levels, these are often still reflected in aspects of the unexpected classes that are found.

In the next section I examine the results of the algorithm on several natural language corpora.

6 Testing the algorithm on real language data

In this section I show how the algorithm performs on several real languages: Samoan, English, French, and Finnish. In all cases the vector embeddings were done using trigram counts and PPMI weighting. I make several simplifying assumptions when dealing with the data in this section: first, I restrict the initial partition of the data to only use the first principal component and to partition this principal component into only two classes. Assuming that the most obvious partition of the full set of sounds is between consonants and vowels, this is equivalent to stipulating that the first partition of a phonological corpus must be into these two categories, and that subclasses must be contained entirely in the set of vowels or the set of consonants. This potentially misses certain classes that span both sets (like the class of $\left[\ \text{+voice}\ \right]$ sounds, or classes containing vowels and glides, such as {i,j} and {u,w}, for example), but greatly reduces the number of classes generated and facilitates interpretation.

Second, I occasionally change the parameter that determines how many principal components of a class are considered. Recall that the default is to cluster only on principal components that account for a greater than average proportion of the variance in the data. I scale this by multiplying it by a factor (so e.g. we might only consider principal components that account for two times the average variance). This is useful because of the varying distributional noise present in real data. All classes returned with a higher threshold will also be returned when the threshold is lowered. I leave the question of whether there is a more principled way to determine this threshold as a topic for future research.

I evaluate the algorithm's performance by inspecting the discovered classes and comparing them to classes that have been proposed by linguists to describe that language's phonology.

6.1 Samoan

The Samoan corpus was generated from a Samoan dictionary (Milner, 1993) and contained 4226 headwords.[13] This is an orthographic representation of Samoan, but there is a close correspondence between orthography and pronunciation. Symbols have been converted to IPA for clarity. Figures 9 to 11 visualise the vector embedding of Samoan.[14]

The retrieved classes are shown in Figure 12. The algorithm was able to successfully distinguish between consonants and vowels. It also makes a rough distinction between long and short vowels, although /a:/ is grouped with the short vowels. Finally, the set of short vowels and /a:/ is split into low and non-low, while the set of long vowels is partitioned and

[13]Thanks to Kie Zuraw for providing this data.

[14]Note that /h/ was successfully classified as a consonant despite appearing somewhat intermediate in the plot. /h/ is a rare phoneme in Samoan, only appearing in loanwords. In the corpus used here, only four words contained /h/, and each of these had /h/ in initial position. This limited the range of available contexts and made the distinction between consonant and vowel less clear than for other sounds.

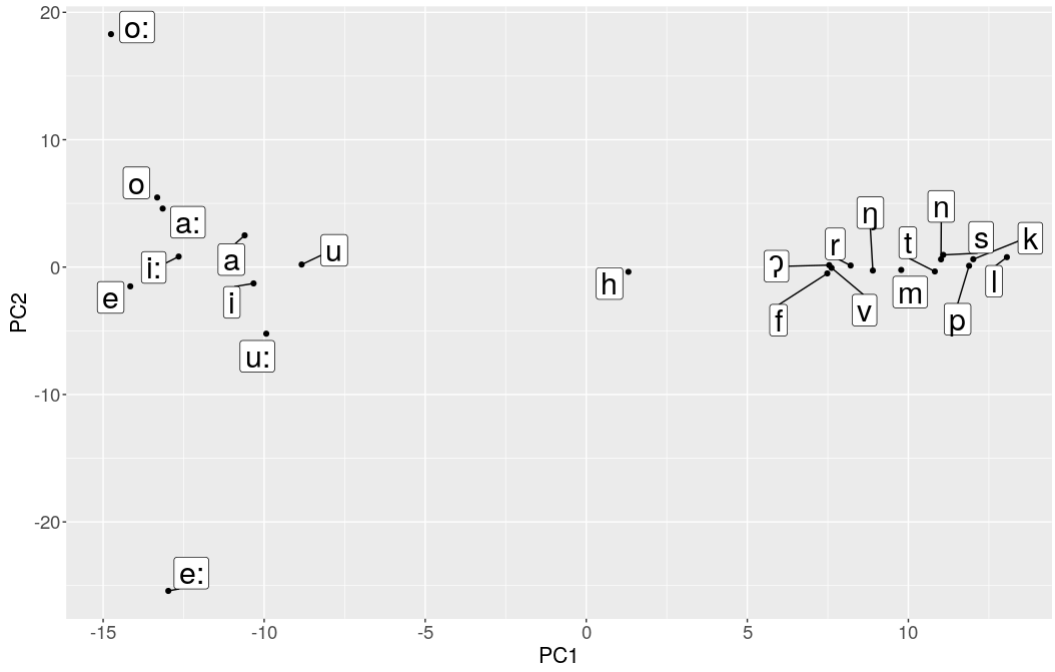

Figure 9: A PCA of the vector embedding of Samoan.

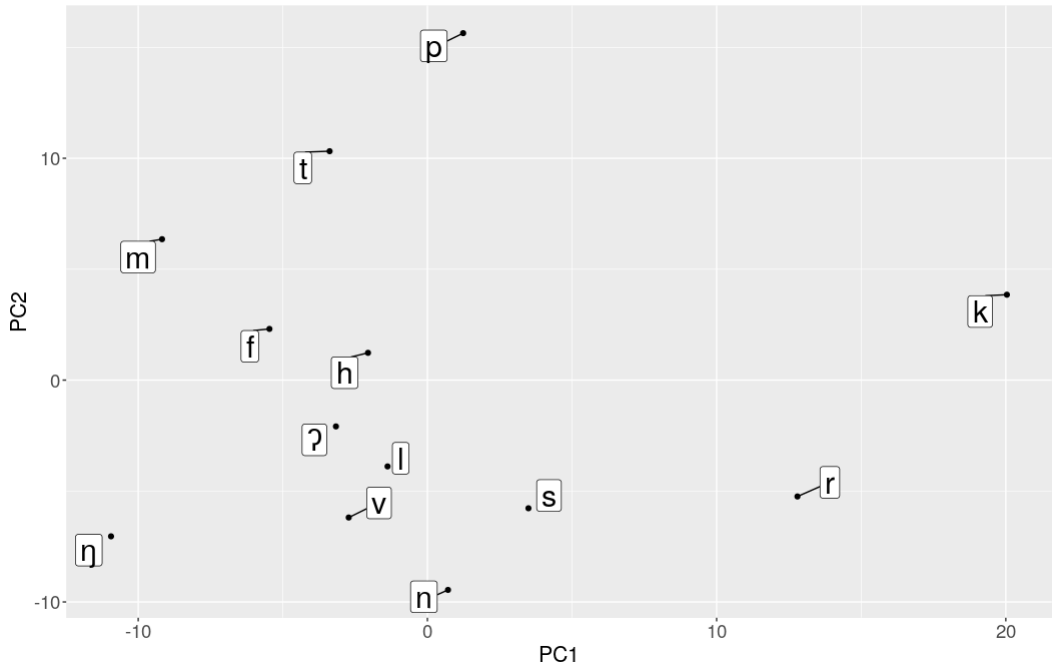

Figure 10: A PCA of the Samoan consonants.

high and mid sets. There does not appear to be sufficient distributional information to make any partitions of the set of consonants. Lowering the variance threshold for which principal components to consider did not result in more classes being learned.

The patterning of /a:/ with the short vowels can be explained by examining its distribu-

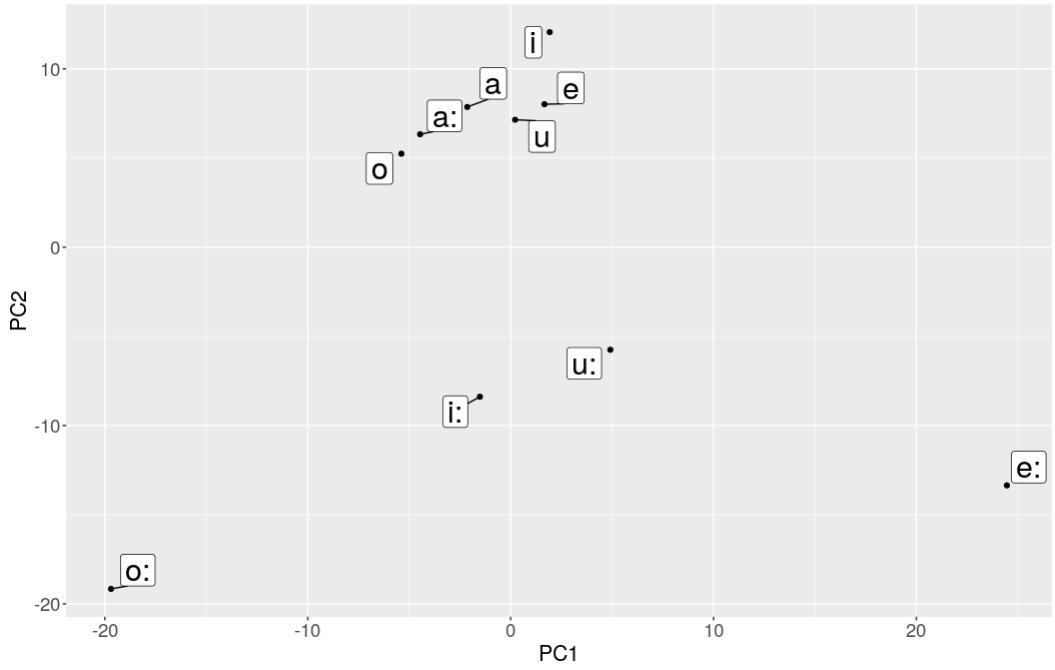

Figure 11: A PCA of the Samoan vowels.

tion. While VV sequences are quite common in Samoan (1808 occurrences in the corpus), VV:, V:V, and V:V: sequences are rarer (226 total occurrences). In 171 of these 226 occurrences, the long vowel is /a:/. Thus /a:/ patterns more like a short vowel than a long vowel with respect to its distribution in vowel sequences, and the algorithm reflects that in its discovered classes. This is an example of a class that cannot be captured using phonetic features, but is valid in the sense that it is salient in the distribution of the language.

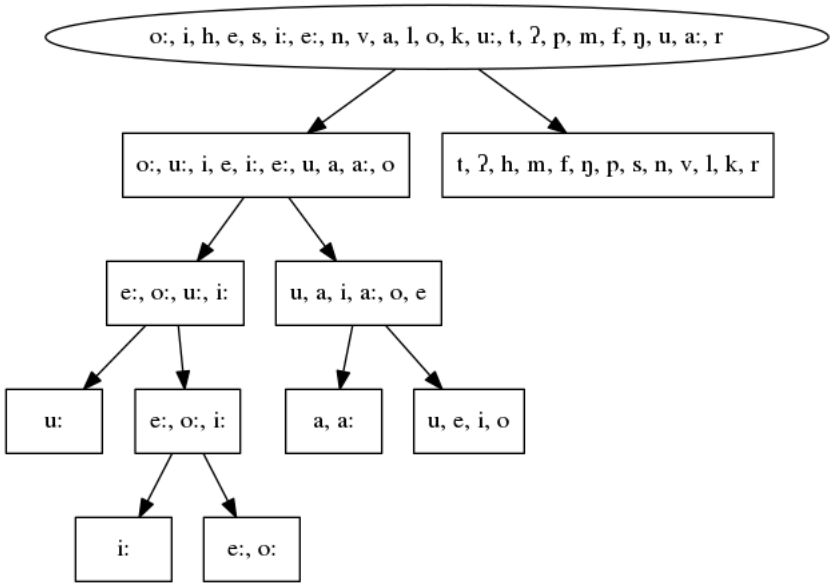

Figure 12: Retrieved classes from Samoan. Arrows indicate parent/child relationships.

To examine whether the trigram window is too small to capture information that might allow the consonants to be grouped, I also ran the algorithm on Samoan with the vowels removed. This should allow the algorithm to better capture any word-level co-occurrence restrictions that might differentiate groups of consonants (e.g. McCarthy, 1986; Coetzee & Pater, 2008). A PCA of the resulting vector embedding of the Samoan consonants is shown in Figure 13.

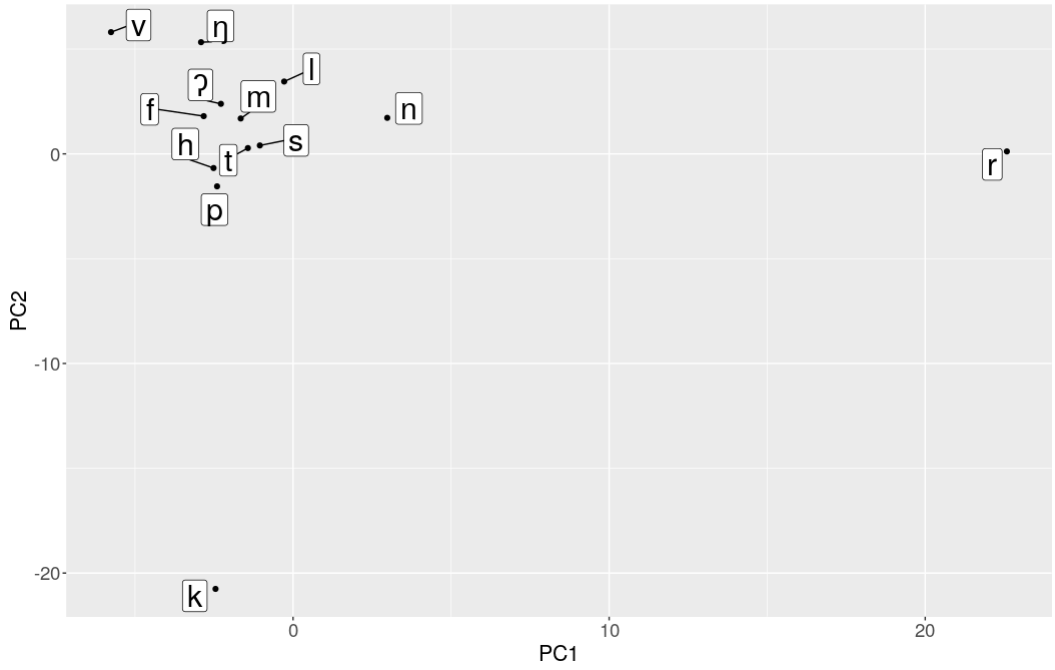

Figure 13: A PCA of the Samoan consonants from a corpus without vowels.

In order to reduce noise, I ran the clustering algorithm on this data with a scaling factor of 1.2 on the variance threshold (i.e. only principal components with at least 1.2 times the average variance were considered). The constraint that the initial partition of the set of sounds must be in two was also removed, because the consonant/vowel distinction is no longer relevant for this data set. This resulted in the classes shown in Figure 14. Here /r/ and /k/ are clearly set apart from the other consonants. These sounds are relatively uncommon in Samoan, being found predominantly in loanwords, and this is reflected in their distribution. Aside from the marginal status of /k/ and /r/ in Samoan phonology, it has hard to justify these classes in a linguistically satisfying way. The additional classes found when the variance threshold was lowered were similarly arbitrary. This suggests that consonant co-occurrence restrictions reflect little more than the special status of /k/ and /r/. Samoan is known to have phonotactic restrictions on root forms (e.g. Alderete & Bradshaw, 2013), and it is possible that running the algorithm on roots rather than headwords would make these patterns more detectable.

Given Samoan's strict (C)V phonotactics, it is perhaps not surprising that distribution yielded few distinctions in the set of consonants. I turn now to English, where the presence of consonant clusters may give us a better chance of retrieving additional phonological information.

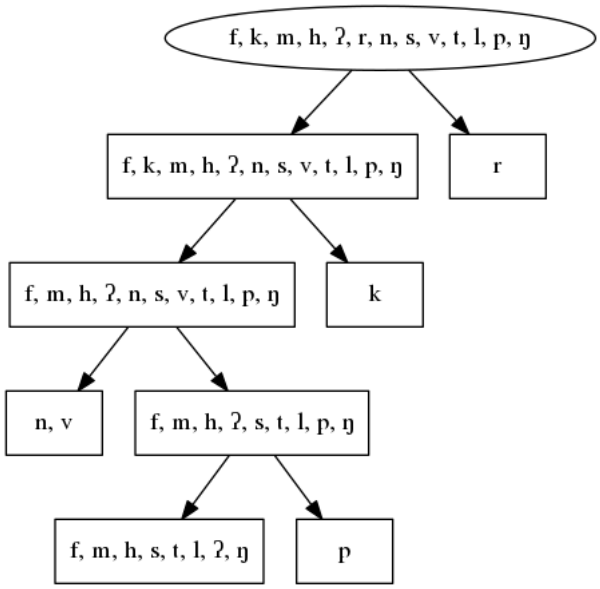

Figure 14: Retrieved classes from Samoan with no vowels. Arrows indicate parent/child relationships.

6.2 English

The English corpus was generated from the CMU pronouncing dictionary,[15] which is phonemically transcribed. Only words with a frequency of at least 1 in the CELEX database were included (Baayen et al., 1995), and some manual error correction was performed.[16] The resulting corpus consisted of 26,552 word types. Figures 15 to 17 visualise the vector embedding of English.

To reduce noise, I ran the clustering algorithm with a scaling factor of 1.1 on the variance threshold. The retrieved classes are shown in Figure 18. The sets of vowels and consonants are correctly retrieved. Within the consonants, there is an eventual distinction betwen the class of coronal obstruents, nasals, and /v/, and all other consonants. The class of velar obstruents {k,g} is recovered, as well as the class of labial obstruents {p,b,f} minus /v/, and the set of labial approximants {r,w}. The vowels are more difficult to interpret, but there are splits that are suggestive of the tense vs. lax distinction.

In a language like Samoan, with a small number of sounds and extremely restricted syllable structure, it is relatively simple to identify the specific distributional properties that lead to a particular class being detected. More phonotactically complex languages like English are not so simple. It would be interesting to investigate what aspects of the distribution led to the detection of the classes found here, but I leave this as a topic for future research.

I turn now to French, a language with similarly complex phonotactics to English.

[15] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[16] See http://linguistics.ucla.edu/people/hayes/EnglishPhonologySearch

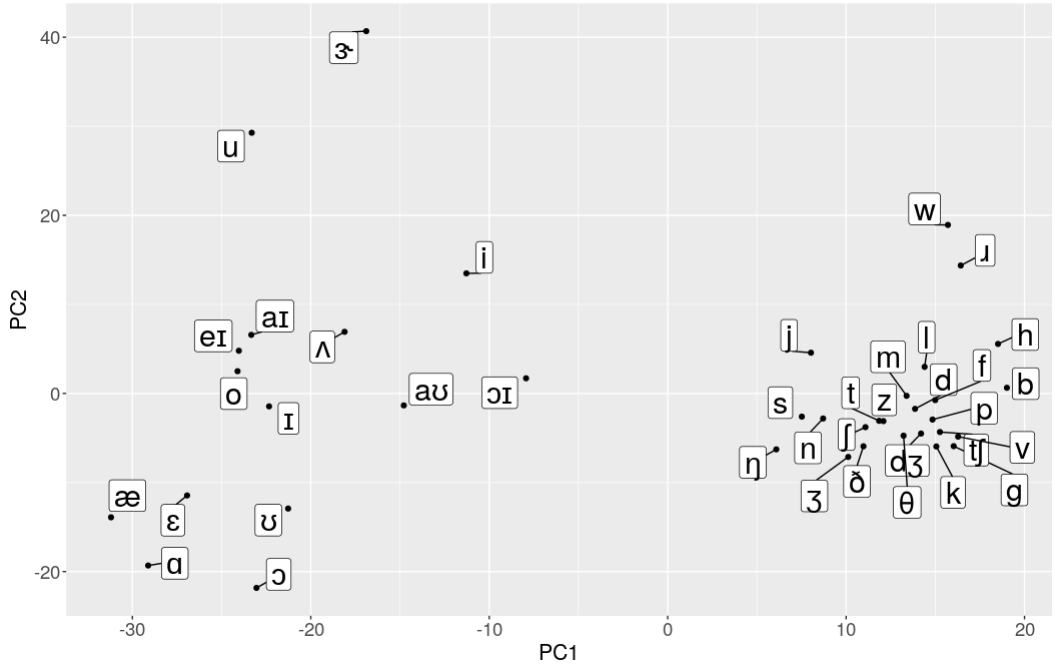

Figure 15: A PCA of the vector embedding of English.

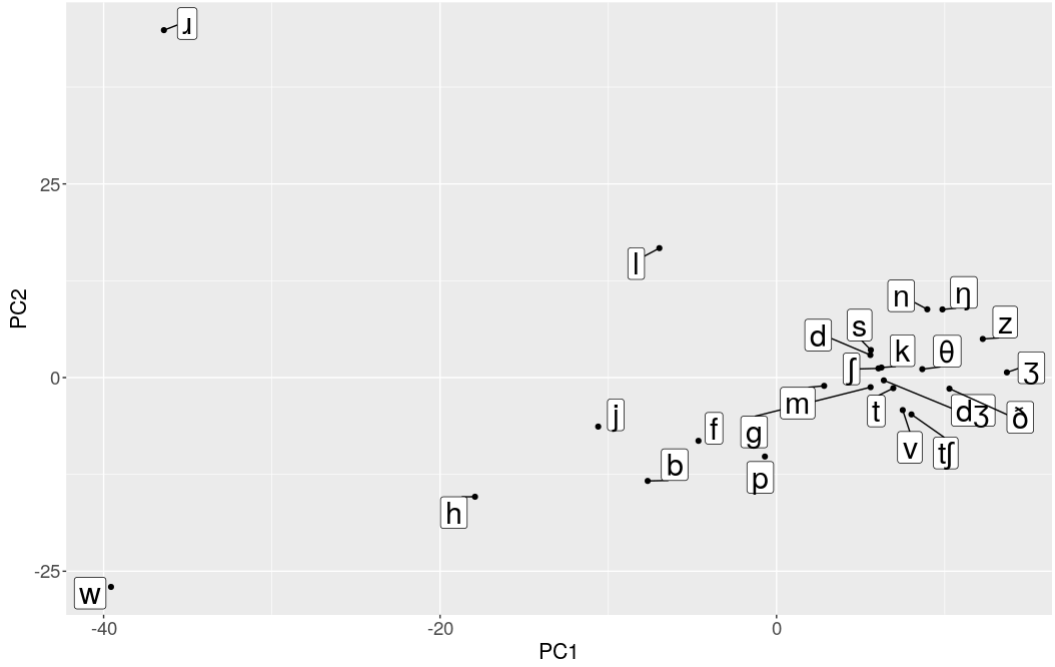

Figure 16: A PCA of English consonants.

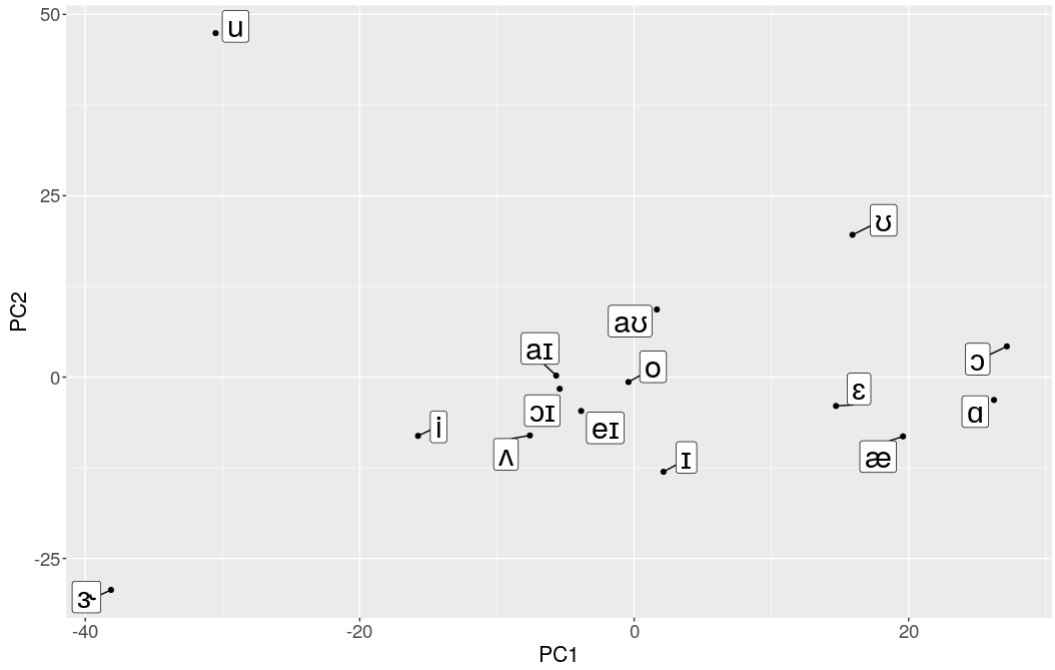

Figure 17: A PCA of English vowels.

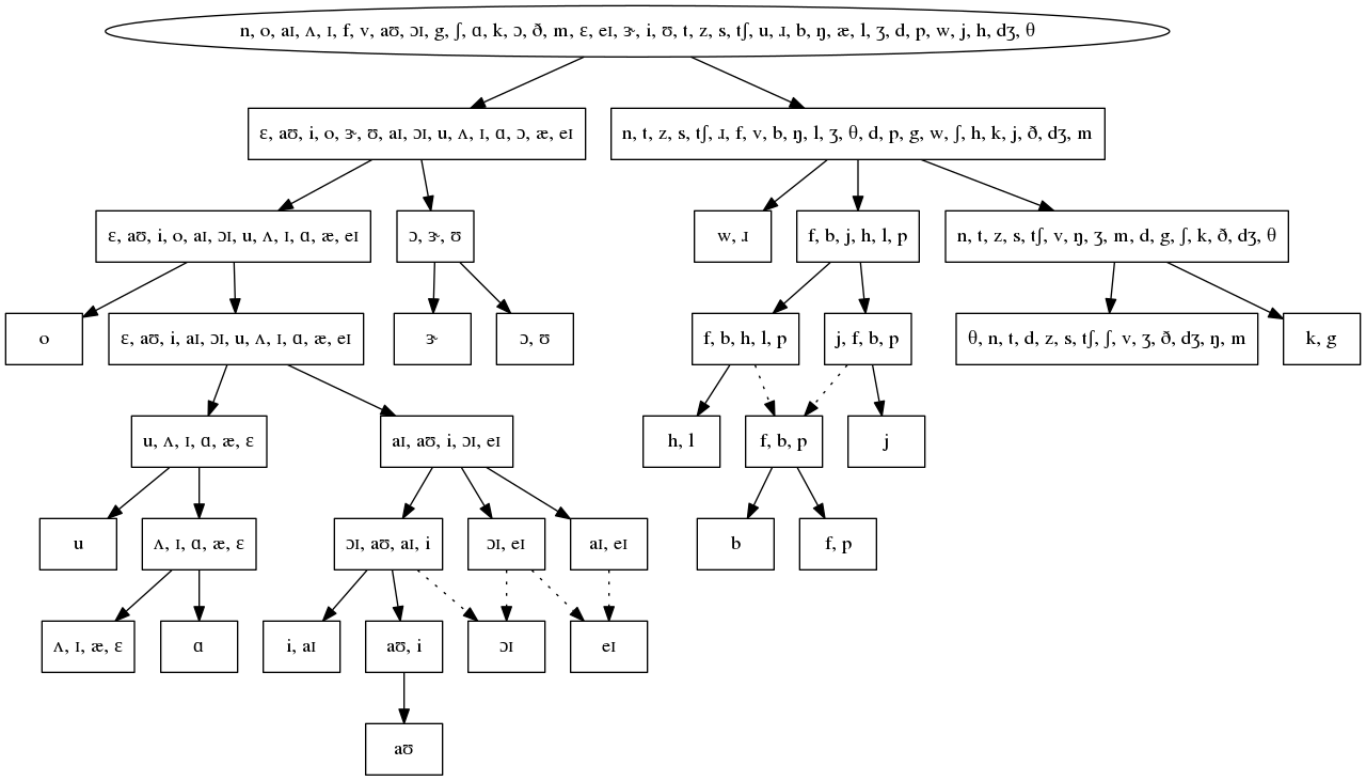

Figure 18: Retrieved classes from English. Arrows indicate parent/child relationships.

6.3 French

The French corpus is the same as the one used in Goldsmith and Xanthos (2009).[17] It consists of 21,768 word types in phonemic transcription. Figures 19 to 21 visualise the vector embedding of French.

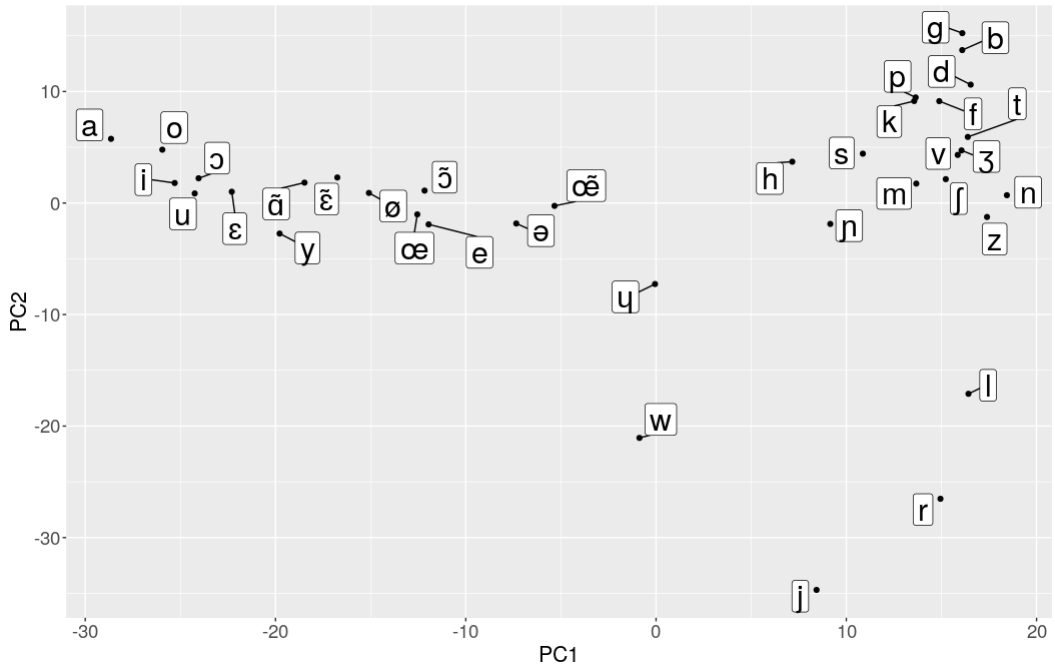

Figure 19: A PCA of the vector embedding of French.

The clustering algorithm was run with a scaling factor of 2 on the variance threshold. The retrieved classes are shown in Figure 22. The sets of consonants and vowels are correctly retrieved. Within the consonants, there is a clean split between approximants and non-approximants, and, within the approximants, between liquids and glides. The glides are further split into rounded and unrounded glides. The vowels are more difficult to interpret, but there is a general split between nasalised vowels and vowels with unmarked roundness on one hand, and the remaining vowels on the other (/y/, /e/, and /ə/ are the exceptions). Again, I leave an enumeration of the distributional properties that characterise these classes as a topic for future research.

[17] Thanks to John Goldsmith for this data.

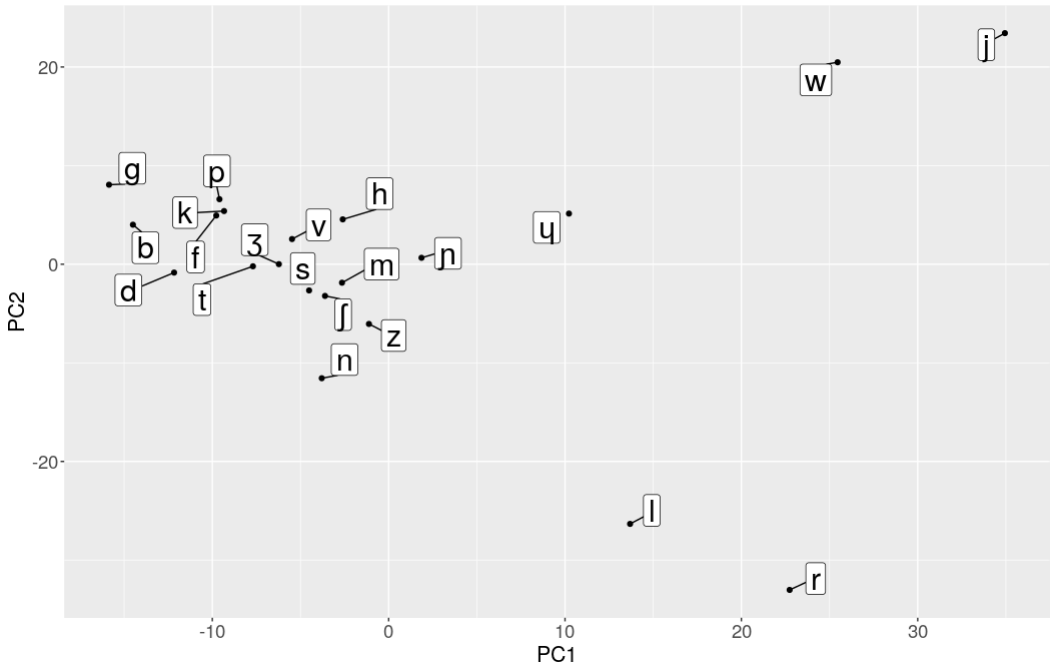

Figure 20: A PCA of French consonants.

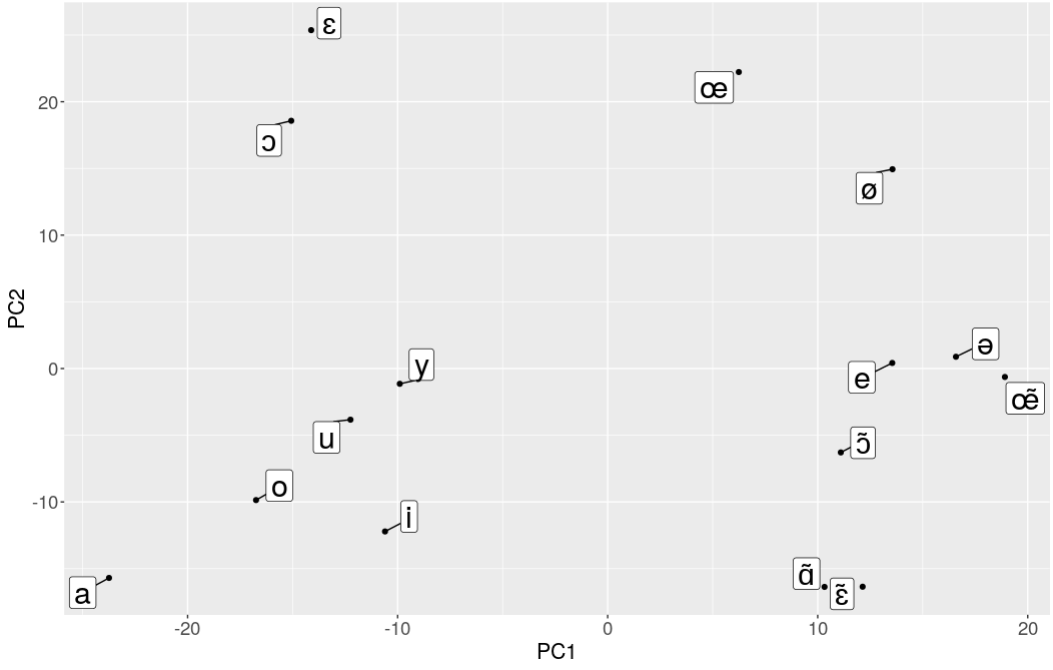

Figure 21: A PCA of French vowels.

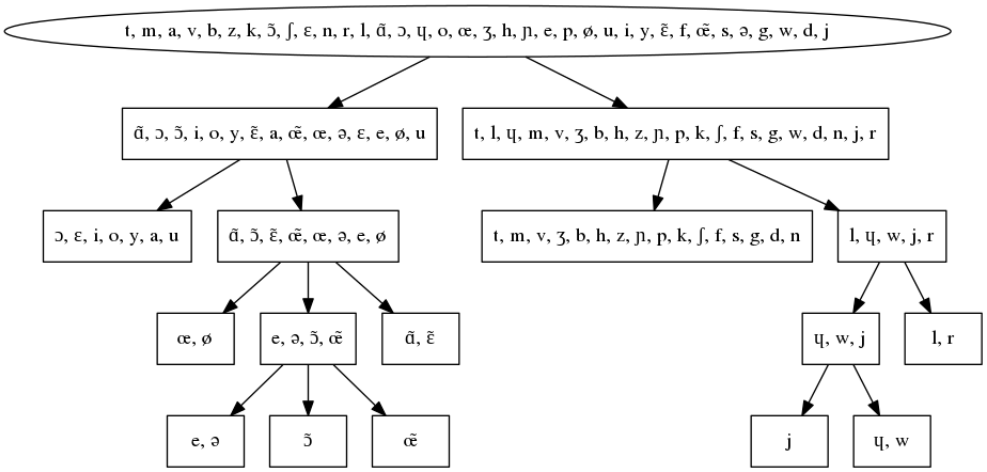

Figure 22: Retrieved classes from French. Arrows indicate parent/child relationships.

6.4 Finnish

Finnish is a central example used in Goldsmith and Xanthos (2009). The Finnish vowel harmony system is sensitive to three classes of vowels: the front harmonizing vowels {y,ö,ä} (IPA: {y,ø,æ}), the back harmonizing vowels {u,o,a}, and the transparent vowels {i,e}. Words tend not to contain both front and back harmonizing vowels, and the transparent vowels can co-occur with either class. Goldsmith and Xanthos show that both spectral clustering and hidden Markov models are able to detect these classes (though see Section 7 for additional discussion).

Because the corpus used in Goldsmith and Xanthos (2009) is not publicly available, I use a corpus generated from a word list published by the Institute for the Languages of Finland.[18] Finnish orthography is, with a few exceptions, basically phonemic, and so a written corpus serves as a useful substitute for a phonemic corpus. Words containing characters that are marginally attested (i.e. primarily used in recent loanwords) were excluded.[19] This resulted in a total of 93,821 word tokens. Long vowels and geminate consonants were represented as VV and CC sequences respectively.

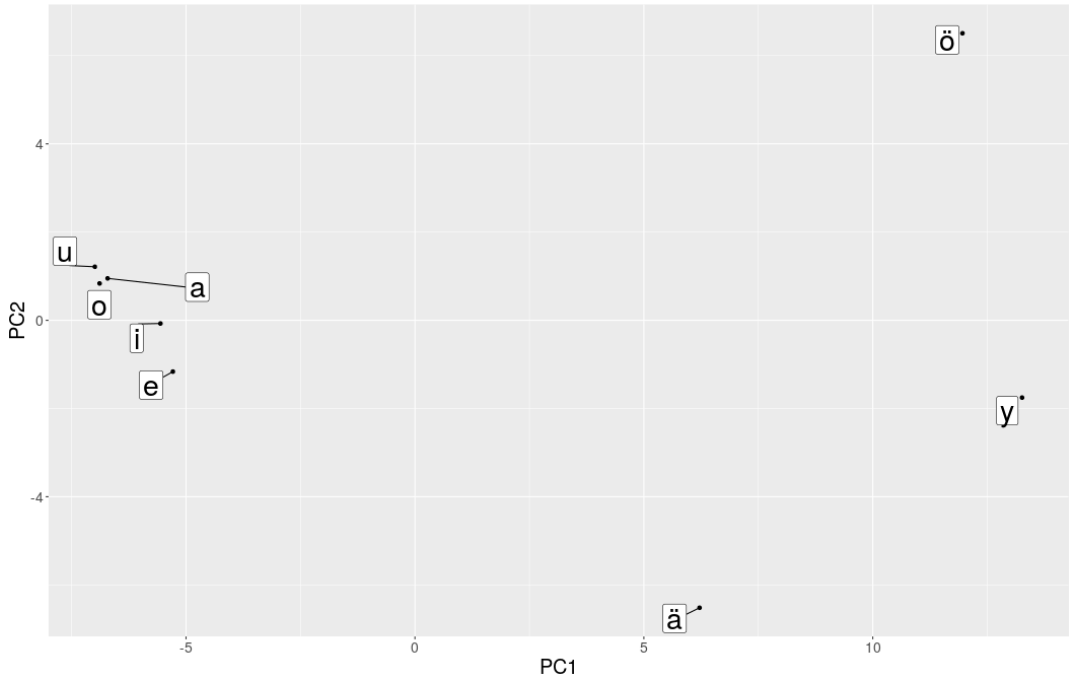

Figure 23: A PCA of the vector embedding of the corpus consisting only of Finnish vowels.

The algorithm was first run on a modified version of the corpus containing only vowels. This mirrors the corpus used in Goldsmith and Xanthos (2009). The vector embedding of this corpus is shown in Figure 23. A scaling factor of 1 was used on the variance threshold, and, as with Samoan consonants, the restriction on the number of classes retrieved in the initial partition was lifted. The retrieved classes are shown in Figure 24. The relevant harmony classes are successfully discovered, and, consistent with the results in Goldsmith and Xanthos

[18] http://kaino.kotus.fi/sanat/nykysuomi/

[19] These characters were c, x, q, z, š, ž, and å.

(2009), the transparent vowels {i,e} pattern more closely with the back vowels than with the front. In addition, classes suggestive of a low/non-low distinction are discovered.

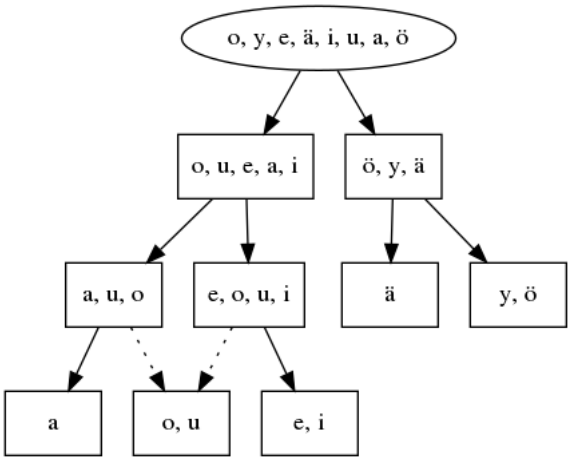

Figure 24: Retrieved classes from the Finnish corpus containing only vowels. Arrows indicate parent/child relationships.

The algorithm was then run on the corpus containing both consonant and vowels. The vector embeddings are shown in Figures 25 to 27.

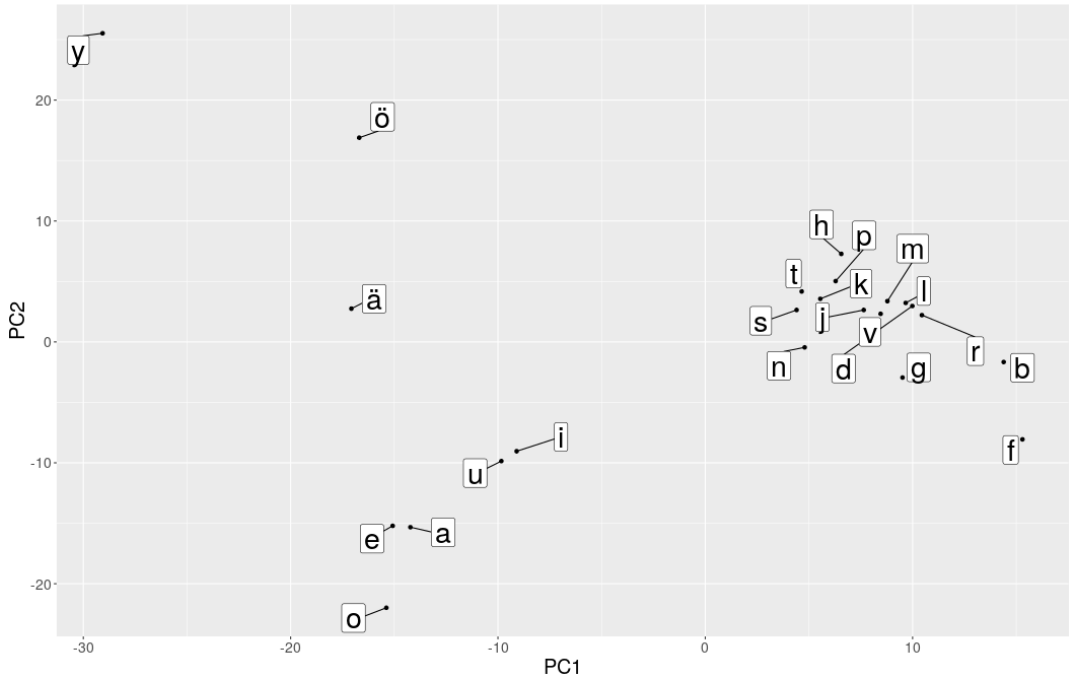

Figure 25: A PCA of the vector embedding of the full Finnish corpus.

The clustering algorithm was run with a scaling factor of 1.2 on the variance threshold. Consonants and vowels were successfully distinguished. Because the focus here is on vowel harmony, and the consonant sub-classes retrieved by the algorithm are not obviously inter-

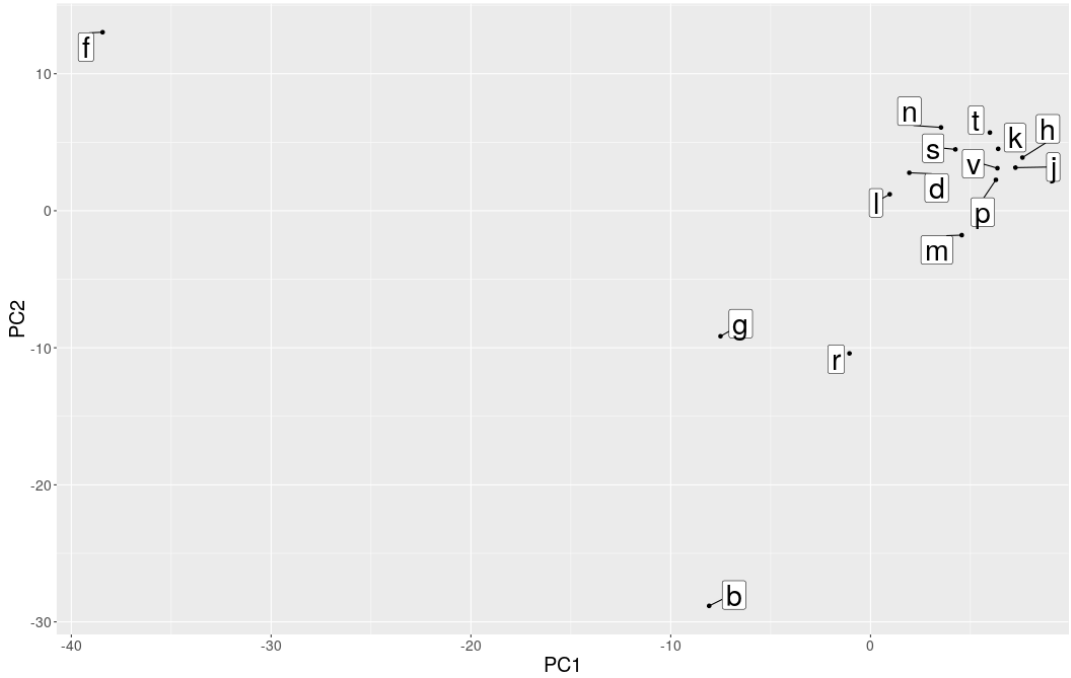

Figure 26: A PCA of the vector embedding of consonants from the full Finnish corpus.

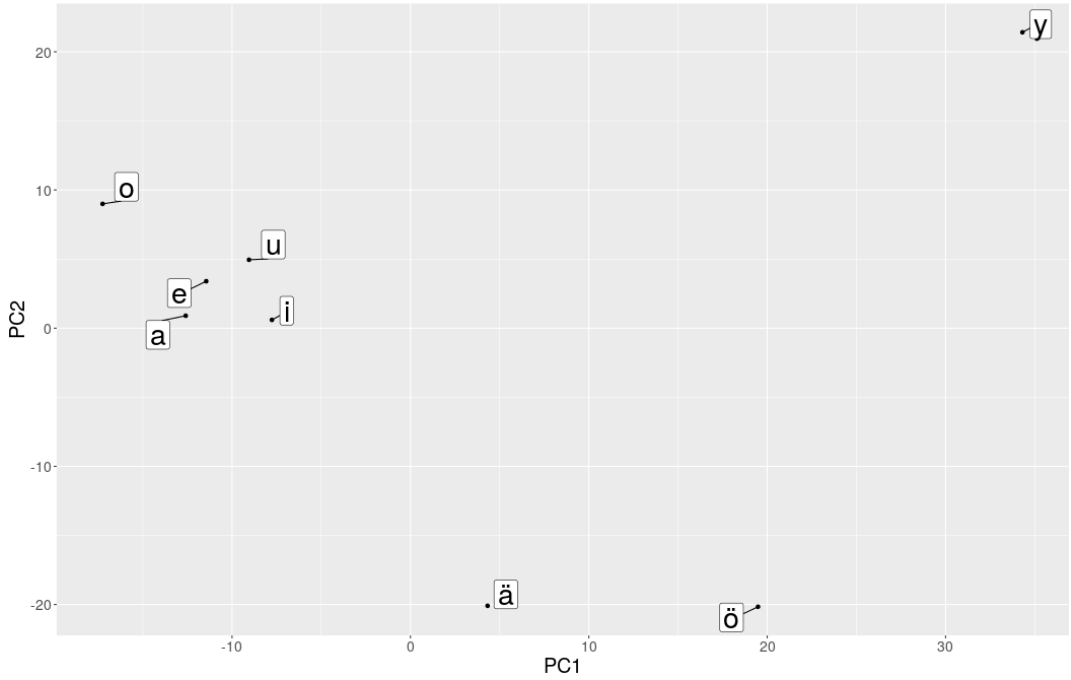

Figure 27: A PCA of the vector embedding of vowels from the full Finnish corpus.

pretable (as Figure 26 suggests), I present only the vowel subclasses here. The retrieved vowel classes are shown in Figure 28.

Here the front harmonizing vowels are differentiated from the transparent and back harmonizing vowels, although the split is not as clean as in the vowel-only corpus: the non-high

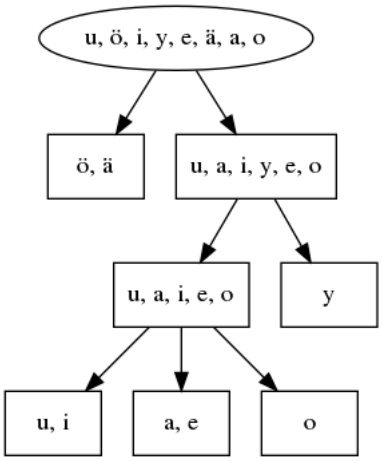

Figure 28: Retrieved vowel classes from the full Finnish corpus. Arrows indicate parent/child relationships.

front harmonizers {ö,ä} form their own class, and only later is {y} split off from the remaining vowels. In addition, the distinction between transparent and back harmonizing vowels is not made, although the set of both is split into classes suggesting a high/non-high contrast. The loss of clear class distinctions when consonants are added back in is a function of the simple trigram counting method used: because Finnish allows consonant clusters, trigrams are not able to capture as much of the vowel co-occurrence as they need to generate the expected classes. More will be said on this point in Section 8.

The algorithm presented here is able to retrieve the correct classes on the corpus containing only vowels, and retrieves classes that capture aspects of the harmony pattern when run on the full corpus. Although the results on the vowel-only corpus seem quite comparable to those in Goldsmith and Xanthos (2009), the next section will discuss why these constitute an improvement in several ways beyond simply the classes that are retrieved.

7 Comparison with past work

A direct comparison of this algorithm to past approaches is difficult because of the lack of a clear quantitative measure of success, the lack of publicly available implementations, and the use of different data sets. Qualitative comparison is possible, however, particularly for the English, French, and Finnish data sets, which are similar or identical to some of those used by Goldsmith and Xanthos (2009) and Calderone (2009). From this perspective, the current algorithm offers several notable improvements.

In all past approaches, except Nazarov (2016), there is no clear method for producing multiple partitions of the same set of sounds (i.e. multiple class membership), and no clear method to partition subsets of the segmental inventory without tailoring the input to include only these subsets. As shown by its application to Parupa, the current algorithm is capable of both these things. Because multiple class membership and privative specification are important properties of most phonological characterizations of a language, these are desirable properties.

The spectral clustering algorithm detailed in Goldsmith and Xanthos (2009) is similar to the current approach in that it decomposes a matrix representation of the distribution of sounds into a simple component that allows groups of sounds to be extracted. There are several aspects in which the current algorithm outperforms spectral clustering. First, spectral clustering is not able to produce an accurate separation of consonants and vowels in any of the languages it is applied to (English, French, and Finnish), although they suggest performance could be improved by considering additional contexts when generating the matrix. The current algorithm was able to produce this separation accurately in all cases tested here. Second, Goldsmith and Xanthos do not provide an explicit method for extracting the optimal number of classes from the component aside from visual inspection.

The maximum entropy hidden Markov model approach, also detailed in Goldsmith and Xanthos (2009), performs better on the consonant and vowel distinction, accurately retrieving it in English and French (Finnish is not presented). Further, it is able to identify vowel classes that participate in harmony processes in Finnish when the input consists only of vowels, and loosely captures a distinction between intervocalic and post-consonantal consonants in French. It also provides a more deterministic method for extracting classes by comparing the emission probabilities of segments in each state, although the translation of these numbers into classes is still essentially the responsibility of the analyst. The algorithm presented here performs at least as well, and, importantly, does not require the number of classes (i.e. states of the hidden Markov model) to be specified in advance, which represents a significant increase in robustness and generalisability.[20]

The independent component analysis method described in Calderone (2009) seems to be able to distinguish between consonants and vowels, as well as suggesting the existence of subclasses within these. Similar to spectral clustering, however, Calderone does not provide a method for determining exactly how many classes are present: evidence for classes comes from visual inspection of the individual components and of self-organizing maps, which use neural networks to generate two-dimensional grid visualizations based on these components (Kohonen, 2002).

A direct comparison with Nazarov (2016) is more difficult. The algorithm presented there uses maximum entropy learning to induce phonotactic constraints from input data (Hayes & Wilson, 2008), and forms classes by grouping segments together that are targeted by constraints in similar contexts. The toy language on which it is tested contains three phonotactic constraints that refer to a single segment (no word-final /m/), one class of segments (no nasals word-initially), and two classes of segments (no labials between high vowels). The algorithm is generally successful in learning constraints that refer to classes of sounds, although less reliably so for the final constraint involving two interacting classes. Two things are worth noting: first, although the toy language employed has strict CVCVC

[20]When looking at natural language data above, I stipulated that the initial partitioning of the segmental inventory should only consider the first principal component, and should consider only a partition of this principal component into two classes. This may strike the reader as similar to specifying the number of classes in advance. The crucial distinction is that this stipulation is not in general *necessary* for producing the consonant/vowel distinction, but is simply used to reduce the number of other partitions of the full inventory that are generated, and to avoid premature division into subclasses of either the consonant or vowel classes. The exception to this is French, where rounded glides are clustered with a subset of the vowels if the first principal component is allowed to be partitioned into three classes.

word forms, the algorithm does not appear to learn a consonant/vowel distinction. It is unclear whether the relevant constraints are simply not reported, or if the increased size of the relevant classes proves problematic (i.e. the number of contexts that must be generalized over is too large). Second, the phonotactic constraints are never violated in the input data, which means it is unclear how well the algorithm performs on more gradient cases. By contrast, the algorithm presented in this paper can learn both large and small classes, and functions reasonably well as noise is added.

8 Discussion and conclusion

The question of how much and what kinds of information about phonological classes can be retrieved from distributional information is of considerable interest to phonological theory. The algorithm described in this paper accurately retrieves the intended classes from an artificial language with a reasonably complex class structure, even in the presence of distributional noise. When applied to real languages, it successfully distinguishes consonants from vowels in all cases investigated here, and can make several interpretable distinctions within these categories, such as a near categorical distinction between long and short vowels in Samoan, and a distinction between glides, liquids, and nasals/obstruents in French.

Although the results may seem modest, they are encouraging considering the paucity of the data. No recourse at all is made to the phonetic properties of the sounds, and the representation of the data is phonemic. Integrating sensitivity to additional information such as allophones and syllable boundaries would likely increase the performance of the algorithm and make it more realistic regarding learning.

In a more fully realised model of phonological learning, a necessary subsequent step would be to derive a feature system from the learned classes. This step is not treated in this paper, but is discussed in Mayer and Daland (submitted), where we show that, given certain assumptions about what kinds of featurisations are allowed, a sufficient feature system is derivable from a set of input classes. These two papers may be seen as complementary, and as potential components of a more realistic model of phonological learnability that takes into account other important sources of information, such as phonetic similarity and alternations.

An additional interesting result here is that distributional information is not equally informative for all classes across all languages. Distributional information produced an interpretable partition of vowels in Samoan, but there was virtually no meaningful structure within the class of consonants, even when vowels were removed from the corpus. Indeed, the phonology of the language (including alternations) might not justify any such structure. French and English, on the other hand, had more interpretable results for consonants, but of the two, the result in French more closely matched a typical linguistic description. This suggests that the phonotactics of any given language may refer only to a limited set of phonological classes, and accordingly that some languages may reflect their phonotactics in their distributions more strongly than others.

This study suggests a variety of possibilities for future research, both in terms of improving the performance of the algorithm and of more broadly exploring the role of distributional learning in phonological acquisition.

A desirable property of the structure of the algorithm presented in this paper is that

it is *modular*, in the sense that the two components, vector embedding and clustering, are essentially independent of one another, and can be modified individually. This structure, first quantifying similarity between sounds and subsequently using clustering to extract classes, provides a useful conceptual framework from which to approach problems of distributional learning in phonology in general, and its modular structure lends itself to exploration and iterative improvement.

The counting method employed in the vector embedding step is almost certainly a source of difficulty in the results presented here. For the case of the artificial language Parupa, trigram counts were sufficient to capture *all* phonological constraints in the language, and the model performed accordingly well. It is likely the case that considering additional aspects of context would improve performance on the real languages, although simply increasing the size of the contexts considered in an n-gram model will lead to data sparsity issues. A particularly interesting possibility would be to perform vector embedding using recurrent neural networks (RNNs), which can generate vector representations of sounds without being explicitly told which features of the context to attend to (e.g. Rodd, 1997; Mikolov et al., 2010; Doucette, 2017). This approach may have limited explanatory value in the sense of obscuring what aspects of the context are important, but could help to provide an upper bound for how much information about phonological classes is present in the context. A potential issue with this approach is that RNNs typically require a large corpus for training, necessitating data beyond the simple word lists used here, and making this method less useful for languages with small or no transcribed corpora beyond simple word lists.

An additional consideration is that this algorithm makes a fairly broad pass over the language. Meaningful distributional information about a class might be present in only very specific contexts, and this information may be indistinguishable from noise and similarly suppressed by PCA. A principled way of attending to specific contexts, perhaps along the lines of Nazarov (2016), has the potential to allow more granular classes to be revealed.

Turning to more general considerations, there are many broad questions about the role of distributional learning that could be addressed by experimental work, particularly artificial grammar learning (AGL) experiments. Substantive bias effects (a preference for learning phonetically coherent classes) are notoriously elusive in such studies (Moreton & Pater, 2012), which seems at odds with the hypothesis that phonological classes in real languages should be phonetically coherent. To investigate the role of distributional learning, researchers might perform studies that investigate whether classes that are both phonetically coherent and highly salient in the distribution of participants' native languages are generalised more robustly in AGL tasks than classes that are just distributionally salient or just phonetically coherent. In addition, it would be interesting to investigate whether distributional learning of phonological classes is a strategy available to infants (similar to word segmentation; e.g. Saffran et al., 1996), or if it is a higher level strategy that does not become available until after further development.

Several current debates in phonology revolve around how great a role distributional learning plays in the acquisition and transmission of phonological structure. The algorithm presented in this paper provides some insight into what kinds of phonological information are salient in distributional data. It is my hope that this might subsequently inform further study of the extent to which human learners are able to integrate this information into their phonological grammars.

References

Aggarwal, C. C., & Reddy, C. K. (2013). *Data clustering: Algorithms and applications*. CRC Press.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*, 716-723.

Alderete, J., & Bradshaw, M. (2013). Samoan root phonotactics: Digging deeper into the data. *Linguistic Discovery*, *11*.

Andrews, M., Vigliocco, G., & Vinson, D. (2009). Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, *116*, 463-498.

Archangeli, D., & Pulleyblank, D. (2015). Phonology without universal grammar. *Frontiers in Psychology*, *6*, 1229.

Baayen, R., Piepenbrock, R., & Gulikers, L. (1995). *CELEX2 LDC96L14.* Web Download. Philadelphia: Linguistic Data Consortium.

Blevins, J. (2004). *Evolutionary phonology: The emergence of sound patterns*. Cambridge: Cambridge University Press.

Bruni, E., Tran, N. K., & Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, *49*, 1-47.

Bybee, J. (1995). Regular morphology and the lexicon. *Language and Cognitive Processes*, *10*.

Calderone, B. (2009). Learning phonological categories by independent component analysis. *Journal of Quantitative Linguistics*, *16*.

Chomsky, N., & Halle, M. (1968). *The sound pattern of English*. New York: Harper & Row.

Church, K. W., & Hanks, P. (1990). Word association, norms, mutual information, and lexicography. *Computational Linguistics*, *16*, 22-29.

Coetzee, A. W., & Pater, J. (2008). Weighted constraints and gradient restrictions on place co-occurrence in Muna and Arabic. *LI*, *26*, 289-337.

Dagan, I., Marcus, S., & Markovitch, S. (1993). Contextual word similarity and estimation from sparse data. In *ACL-93* (p. 164-171). Columbus, Ohio.

Doucette, A. (2017). Inherent biases of recurrent neural networks for phonological assimilation and dissimilation. In *Proceedings of the 7th workshop on cognitive modeling and computational linguistics (CMCL 2017)* (p. 35-40).

Ellison, T. M. (1991). The iterative learning of phonological constraints. *Computational Linguistics*, *20*.

Ellison, T. M. (1994). *The machine learning of phonological structure* (Unpublished doctoral dissertation). University of Western Australia.

Everitt, B. S., & Dunn, G. (2001). *Applied multivariate data analysis* (2nd ed.). London: Arnold.

Fano, R. M. (1961). *Transmission of information: A statistical theory of communications*. MIT Press.

Goldsmith, J. (2010). Segmentation and morphology. In A. Clark, C. Fox, & S. Lappin (Eds.), *The handbook of computational linguistics and natural language processing* (p. 364-393). Wiley Blackwell.

Goldsmith, J., & Xanthos, A. (2008). Three models for learning phonological categories. Technical report 2008-8. Chicago: Department of Computer Science, University of

Chicago.
Goldsmith, J., & Xanthos, A. (2009). Learning phonological categories. *Lg*, *85*, 4-38.
Goldwater, S., Griffiths, T. L., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, *112*, 21-54.
Hayes, B., & Wilson, C. (2008). A maximum entropy model of phonotactics and phonotactic learning. *LI*, *39*(3), 379 - 440.
Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*.
Jurafsky, D., & Martin, J. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech processing.* Upper Saddle River, NJ: Prentice-Hall.
Kaiser, H. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*, 187-200.
Kaisse, E. M. (2002). *Laterals are [-continuant].* MS, University of Washington.
Kiparsky, P. (1973). Phonological representations. In O. Fujimura (Ed.), *Three dimensions of linguistic theory* (p. 1-136). Tokyo: TEC Co.
Kohonen, T. (2002). *Self-organizing maps.* Heidelberg: Springer-Verlag.
Labov, W., Ash, S., & Boberg, C. (2006). *The atlas of North American English.* Berlin: Mouton de Gruyter.
Lin, Y. (2005). *Learning features and segments from waveforms: A statistical model of early phonological acquisiton* (Unpublished doctoral dissertation). UCLA.
MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematics, statistics, and probability* (Vol. 1, p. 281-296).
MacWhinney, B., & O'Grady, W. (Eds.). (2015). *The handbook of language emergence.* Chichester: John Wiley & Sons.
Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing.* Cambridge, MA: MIT Press.
Mayer, C., & Daland, R. (submitted). A method for projecting features from observed sets of phonological classes.
McCarthy, J. J. (1986). OCP effects: Gemination and antigemination. *LI*, *17*, 207-263.
Mielke, J. (2008). *The emergence of distinctive features.* Oxford: Oxford University Press.
Mielke, J. (2012). A phonetically-based metric of sound similarity. *Lingua*, *1222*, 145-163.
Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khundanpur, S. (2010). Recurrent neural network based language model. In *Interspeech* (p. 1045-1048).
Milner, G. (1993). *Samoan dictionary: Samoan-English, English-Samoan.* Polynesian Press.
Moreton, E., & Pater, J. (2012). Structure and substance in artificial-phonology learning. part i: Structure, part ii: Substance. *Language and Linguistics Compass*, *6*, 686-701 and 702-718.
Müller, E., Günnemann, S., Assent, I., & Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of VLDB '09*.
Nazarov, A. I. (2016). *Extending hidden structure learning: Features, opacity, and exceptions* (Unpublished doctoral dissertation). University of Massachusetts Amherst.
Niwa, Y., & Nitta, Y. (1994). Co-occurence vectors from corpora vs. distance vectors from dictionaries. In *ACL-94* (p. 304-309).

Peperkamp, S., Le Calvez, R., Nadal, J., & Dupoux, E. (2006). The acquisition of allophonic rules: Statistical learning with linguistic constraints. *Cognition*, *101*, B31-B41.

Powers, D. M. W. (1997). Unsupervised learning of linguistic structure: An empirical evaluation. *International Journal of Corpus Linguistics*, *2*, 91-132.

Redington, M., Chater, N., & Finch, S. (1998). Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, *22*, 425-469.

Rodd, J. (1997). Recurrent neural-network learning of phonological regularities in Turkish. In T. M. Ellison (Ed.), *CoNLL97: Computational natural language learning, ACL* (p. 97-106).

Saffran, J. R., Newport, E. L., & Aslin, R. N. (1996). Word segmentation: The role of distributional cues. *Journal of Memory and Language*, *35*, 606-621.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461-464.

Vennemann, T. (1974). Sanskrit *ruki* and the concept of a natural class. *Linguistics*, *130*, 91-97.

Wang, H., & Song, M. (2011). Ckmeans.1d.dp: Optimal k-means clustering in one dimension by dynamic programming. *The R Journal*, *3*, 29-33.

Wonnacott, E., Newport, E., & Tanenhaus, M. (2008). Acquiring and processing verb argument structure. *Congitive Psychology*, *56*, 165-209.