Vowel Harmony

Statistical Methods for Linguistic Analysis

Rebecca Knowles
Haverford College
Academic Year 2011-2012

Senior Linguistics Thesis (at Swarthmore College)

CONTENTS

VOWEL HARMONY[1]

STATISTICAL METHODS FOR LINGUISTIC ANALYSIS

ABSTRACT

Vowel harmony, a phonological pattern in which vowels within a given domain are required to agree in properties such as tongue position or lip rounding, is a fascinating and fairly widespread phenomenon in the world's languages. Languages vary in their vowel harmony typologies, as well as the extent to which vowel harmony as a phonological constraint is violable. Simple statistical methods can capture interesting facets of vowel harmony systems, as well as provide a way of quantifying vowel harmony so that harmonic systems can be compared. This thesis aims to compare a number of statistical machine learning and natural language processing methods for vowel harmony, culminating in the presentation of a unified tool for visualizing and "diagnosing" vowel harmony systems from data in an unsupervised manner.

1 INTRODUCTION

This thesis explores statistical models and related visualization tools for understanding vowel harmony. While vowel harmony is attested in many languages, it varies by language with respect to which phonological features are shared within words as well as the extent to which disharmony (deviations from the constraint) occurs. These features make it an excellent candidate for statistical natural language processing and machine learning approaches.

The ability to quantify vowel harmony opens up a number of opportunities for research, including the ability (given the necessary corpora) to trace change in a language's vowel harmony system diachronically or to explore or even discover small but statistically significant levels of harmony. Unsupervised models in particular offer useful ways of quickly communicating information about vowel harmony systems while requiring minimal knowledge about the language in question.

[1] Thanks to my thesis advisor, Nathan Sanders, for his support and feedback and to my Linguistics major advisor, K. David Harrison, for introducing me to such an interesting topic. Thanks to my second faculty reader, Kevin Ross, for his questions and comments. Thanks to my fellow NLP enthusiasts and members of the NLP/CL first reader triangle – Kristen Allen and Jon Gluck – for good ideas and camaraderie. Thanks to my second student reader Andrew Cheng for providing a fresh perspective. To all my family, friends, teachers, and professors who helped along the way – thank you!

I begin with an introduction to vowel harmony, discuss important features of relevant data and computational methods, and describe methods for statistically modeling and quantifying vowel harmony. Additionally, I introduce a simple model closely related to Hidden Markov Models as well as a visualization tool that can be used for the output of both the newly introduced model and Hidden Markov Models. The thesis culminates with results of experiments on six languages, a comparison of two models, and a discussion of future work.

2 Vowel Harmony

Vowel harmony is a phonological pattern in which vowels within some domain – typically the word – share one or more phonological features, like lip rounding or tongue position (Katamba, 1989: 211). This is considered a long-distance process, since vowels can harmonize across intervening consonants and even certain non-harmonizing vowels. Presented below are the main features of vowel harmony systems, with examples from several languages. I examine basic harmony systems, the different roles vowels can play in harmony, and some particular theoretical concerns that relate directly to the use of computational methods.

2.1 Introduction to Vowel Harmony Systems

A good place to begin is with the case of Finnish, a Uralic language known to be harmonic. The vowel inventory for Finnish contains eight vowels (Table 1).

Table 1: Finnish Vowels

Front		***Back***		
Unrounded	Rounded	Unrounded	Rounded	
i	y		u	***High***
e	ö		o	***Mid***
ä		a		***Low***

Orthographic <ä> and <ö> correspond to IPA /æ/ and /ø/, respectively. Since there is a close correspondence between Finnish orthography and pronunciation, the orthography can be used here without the loss of significant phonological information.

Finnish exhibits *palatal harmony*, dividing words into two classes in terms of the backness of their vowels. That is, all of the vowels in a given word are expected to be either front or back. Onewould expect to see words like *pöytä* ’table’, in which all the vowels are front vowels, or words like *pouta* ‘fine weather’, in which all the vowels are back vowels, but not a word that contained both <o> and <ä>. This phenomenon extends to suffixes as well, which have front and back allomorphs that are selected to match the vowels in the word stem. For example, the suffix *-sta/-stä* varies in accordance with the backness of the word, taking on the form *-stä* with front words and the form *-sta* with back words (Table 2).

Table 2: Palatal Harmony in Finnish

Front words		***Back words***	
väkkärä	‘pinwheel’	makkara	‘sausage’
pöytä	‘table’	pouta	‘fine weather’
käyrä	‘curve’	kaura	‘oats’
tyhmä-stä	‘stupid’ (ill.)	tuhma-sta	‘naughty’ (ill.)

(Hulst and Weijer, 1995: 498)

Palatal harmony is not the only type of harmony that can be observed in the world’s languages. Swahili, along with other Bantu languages, exhibits vowel harmony with respect to height. It has a vowel inventory consisting of five vowels (Table 3) whose orthographic representations correspond to their representations in IPA (Ladefoged, 2005:26).

Table 3: Swahili Vowels

Front		***Back***	
i		u	***High***
e		o	***Mid***
	a		***Low***

While languages with palatal harmony divide the vowels into classes based on their backness, languages with *height harmony* do so based on vowel height (Table 4). Thus in a language like Swahili, it is unsurprising to see vowel pairs like <u> and <i> or <o> and <i> co-occur.

Table 4: Height Harmony in Swahili

	Verb Root		***Suffixed Form***	
High	-ruk-	‘jump, fly’	-ruki-	‘jump at, fly at’
	-andik-	‘write’	-andiki-	‘write for’
Non-high	-som-	‘study’	-some-	‘study for’
	-end-	‘go’	-ende-	‘go for/to/toward’

(Childs, 2003: 70)

It may be noted that not all the roots in Table 4 harmonize fully, as demonstrated by the disharmony in *-andik-* 'write'; since <a> is a low vowel and <i> is a high vowel, their presence together in one verb root appears to violate the height harmony pattern that Swahili is known to exhibit. This is due to a feature of harmony discussed (with Finnish data) in Section 2.3.

2.2 DISHARMONY

I return to the examples from Finnish in order to discuss the problem of the violability of vowel harmony constraints. While most Finnish words exhibit whole-word harmony, not all do. Disharmonic words – words that violate harmony constraints – can appear in otherwise harmonic languages for several reasons. Loanwords can be a source of disharmony in harmonic languages. This is the case for some loanwords in Finnish (Table 5), like *tyranny* 'tyrant', which contains front vowels <y> and <i> and the back vowel <a>, making it disharmonic with respect to backness.

Table 5: Finnish Loanwords

Loanword	
vúlgääri	'vulgar'
týranni	'tyrant'

(Ringen and Heinämäki, 1999: 306)

Words from the lexicon can also be disharmonic for other reasons such as consonant blocking (discussed in Section 2.7), but in harmonic languages, such words are naturally an exception. For more examples of Finnish disharmony and analysis thereof, see Ringen and Heinämäki (1999).

Turkish – which exhibits palatal harmony as well as roundness harmony (see Section 2.4) – is a language known to have a number of disharmonic roots (Table 6), despite being a commonly used example of a language with vowel harmony. An example of this is the word *muzip* 'mischievous', which has the back vowel <u> and the front vowel <i>. Clements and Sezer discuss the theoretical implications of disharmony, noting that choosing to consider loanwords exceptions to harmony based on their being nonnative is problematic due the fact that loanwords commonly adapt to the phonological rules of the language into which they have been borrowed (1982: 226).

Table 6: Disharmonic Words in Turkish

Disharmonic Word[2]	
muzip	'mischievous'
anne	'mother'
peron	'railway platform'
hani	'where is'

(Clements and Sezer, 1982: 222)

As seen in both Finnish and Swahili, some affixes alternate in order to harmonize with stems. While this is also the case with many suffixes in Turkish, Turkish also exhibits disharmony due to some disharmonic suffixes. These are suffixes, like *-edur/-adur* 'verb-forming' in which one or more of the vowels in the suffix does not alternate in accordance with the harmony rules. In the case of *-edur/-adur*, the first vowel harmonizes, but the second does not. Comparing *gid-edur-sun* 'let him keep going' and *bak-adur-sun* 'let him keep looking' shows that the final vowel <u> of the suffix does not harmonize, but the initial vowel <a> or <e> alternates in accordance with the backness of the backness of the word (Table 7).

Table 7: Disharmonic Suffixes in Turkish

Disharmonic Word	
gid-edur-sun	'let him keep going'
bak-adur-sun	'let him keep looking'

(Clements and Sezer, 1982: 231)

Whether disharmonic words in Turkish are disharmonic due to loanword status, disharmonic affixes, or to other intricacies of the rules governing harmony, the concern in taking a computational approach to vowel harmony is simply that they exist even in languages considered to be harmonic. Thus the methods used must be flexible enough to take that into account.

2.3 Neutral Vowels

Not every word that appears disharmonic at first glance should be considered disharmonic. Some words that look to be disharmonic in Finnish (Table 8) actually display an important feature of the language's vowel harmony: the *neutral* vowels /e/ and /i/.

[2] The vowels here correspond to the Finnish vowels introduced in Section 2.1.

Table 8: Neutral Vowels in Finnish

Front words		***Back words***	
värttinä	'spinning wheel'	palttina	'linen cloth'
kesy	'tame'	verho	'curtain'

(Hulst and Weijer, 1995: 498)

They are referred to as *neutral* vowels because they have no corresponding back unrounded vowels in the language's vowel inventory. More specifically, this type of neutral vowel is called *transparent*, meaning that the harmonic features spread through these vowels. Very simply speaking, this means that the harmonic feature (e.g. backness) of the vowels on either side of a transparent vowel should match those of the vowels in the word, ignoring the transparent vowel between them. In the Finnish example, the transparent neutral vowels /e/ and /i/ occur in both front words (like *värttinä* 'spinning wheel') and back words (like *palttina* 'linen cloth') without playing a role in harmony. The transparent vowels should be considered separate from the harmonic system, and are essentially ignored – their appearance in a word does not render it disharmonic even if their features would typically cause that to be the case.

While Finnish only exhibits transparent neutral vowels, there is another type of neutral vowel: the *opaque* vowel, which blocks the harmonic process and begins "a new harmonic domain with [its] own feature specification" (Krämer, 2003: 27). In Shona, a language that harmonizes with respect to height and roundness, low vowels block harmony. Shona height harmony is exhibited in *bvum-isa* 'make agree', where the suffix *-isa/-esa* takes its high form (with <i>) to harmonize with the high vowel <u> (Table 9).

Table 9: Height Harmony in Shona

Height Harmonic Shona Words			
oma	'be dry'	om-esa	'cause to get dry'
bvuma	'agree'	bvum-isa	'make agree'

(Beckman, 1997: 1)

The blocking in Shona "neither trigger[s] nor propagate[s] height harmony" (Beckman 1997: 1), instead, only the vowels <i> and <u> are allowed to follow a low vowel. Mid vowels cannot spread across the opaque vowel <a>. For example, the suffix *-isa/-esa* uses its high form in *cheyam-isa*

’make be twisted’; even though the initial vowel in the stem is <e>, a mid vowel, the harmony does not spread across the vowel <a> (Table 10). Instead, the suffix *-isa* is used, since <i> is allowed to follow <a>, while <e> is not.

Table 10: Opaque /a/ in Shona

Shona Blocked Harmony			
shamba	‘wash’	shamb-isa	‘make wash’
cheyama	‘be twisted’	cheyam-isa	‘make be twisted’

(Beckman, 1997: 2)

This is, of course, only one example of opaque vowel blocking. It is also possible for opaque vowels to propagate their phonological features through the word.

Neutral vowels are typically vowels that do not have corresponding vowels in the opposite harmonic class. Examples include the front unrounded vowels in Finnish which have no back unrounded analogs and /a/ in Swahili, which has no high analog.

2.4 Vowel Harmony in Multiple Dimensions

The vowel harmony system in Turkish works in two dimensions – there is palatal harmony as well as labial harmony (or roundness harmony), though the labial harmony is limited to high vowels (Clements and Sezer, 1982: 216). The vowel inventory of Turkish contains eight vowels (Table 11), whose orthography and pronunciation are sufficiently closely related that I present them here in the orthographic forms.

Table 11: Turkish Vowels

Front		***Back***		
Unrounded	Rounded	Unrounded	Rounded	
i	ü	ɨ	u	***High***
e	ö		o	***Mid***
		a		***Low***

(Baker, 2009: 8)

In suffixes that harmonize, it is the case that all high vowels harmonize with respect to roundness and backness, while non-high vowels only harmonize with respect to backness. This can be seen in the genitive singular suffixed form of ‘rope’, *ip-in*, and the genitive singular suffixed form of ‘end’, *son-un*. Here the suffix alternates based on both the backness and the roundness of the vowel in the root. The vowel <i>, appearing in both the root and the suffix, is front unrounded; the vowels <o> and <u> are both back and rounded (Table 12). For non-high suffixes, agreement is confined to backness harmony, as seen in the example *pul-lar*, the nominative plural 'stamp'. In this form the vowels <u> and <a> appear. While they agree with respect to backness (both are back vowels), they disagree with respect to roundedness – <u> is rounded and <a> is not.

Table 12: Turkish Harmony

	Noun Root		***Suffixed Form***	
High Vowel in Suffix	son	'end'	son-un	*Gen. Sg.*
	ip	'rope'	ip-in	*Gen. Sg.*
Non-high Vowel in Suffix	pul	'stamp'	pul-lar	*Nom. Pl.*
	yüz	'face'	yüz-ḽer	*Nom. Pl.*

(Clements and Sezer, 1982: 216)

This phenomenon in Turkish complicates the study of vowel harmony because some vowels participate in only one type of harmony while the others participate in both. The computational approaches in this paper are mainly focused on finding harmony systems in only one dimension, though I do discuss some ways that the models can accommodate multi-dimensional harmony systems.

2.5 More Types of Vowel Harmony

It is clear from these two examples that there are different ways for languages to exhibit vowel harmony. They can exhibit vowel harmony in one dimension, like Finnish, or multiple dimensions simultaneously, like Turkish. Additionally, as in the case of Turkish, it is possible for a type of harmony to be restricted to a subset of the language's vowel space. It should come as no surprise that other types of harmony exist as well, given the examples of height harmony in Shona and Swahili. Additionally, Tangale – a Chadic language spoken in Nigeria (Ethnologue, 2009) – exhibits harmony with an open-close or high-low distinction, which can also be analyzed in terms of the feature [±ATR] (Hulst and Weijer, 1995: 509-510).

In addition to knowing what features can harmonize, it can be important to understand how vowel harmony systems function. This can be framed in terms of a distinction between *stem-*

controlled systems and *dominant-recessive* systems. In the former, stems determine how affixes harmonize. In the latter, there is a given harmonic feature that "dominates" or takes precedence regardless of whether it appears in a stem or an affix (Baković, 2000: ii). Languages with stem-controlled harmony tend to add affixes in only one direction, and the direction of harmony typically matches that (Krämer, 2003: 113), though Clements notes that harmony can be bidirectional – causing both prefixes and affixes to harmonize with the stem (1977). As I intend to explore automatic approaches to finding vowel harmony, it is important to consider whether computational methods will be able to find both types of harmony. Additionally, some methods assume a certain direction of vowel harmony, which could make them less effective at spotting a wide range of vowel harmony systems.

2.6 The Harmonic Domain

The grammatical or morphological word is not necessarily the ideal domain in which to understand vowel harmony, due to cases like compound words which may not exhibit vowel harmony as a unit but would do so individually. It is based on evidence from compound words and Hungarian vowel harmony that Hall (1999: 3) makes the argument for the prosodic (or phonological) word as the domain for vowel harmony and other phonological rules. It is therefore a more appropriate harmonic domain than the grammatical word, but this analysis also has faults. Criticism of the prosodic word as the harmonic domain is based on a concern that vowel harmony does not fit well with other processes that take place with the prosodic word as their domain, since there can be disharmony within prosodic words, no distinction is made between affixes that cohere to the prosodic word and non-cohering affixes with regard to harmonizing, and vowel harmony is typically "obligatory" rather than "optional" (Hulst and Weijer, 1995: 501-502). There can be cases, though, where the domain for vowel harmony is the prosodic word. Due to issues in data availability and formats, which are discussed later, most of these computational methods are forced to operate on the grammatical word. While this is not ideal, the probabilistic nature of the methods should allow for flexibility when it comes to problems like compound words. If there is a particular concern about a dataset containing a large number of compound words, there are various algorithms (ranging from unsupervised to supervised) that could be used for word segmentation in a data preprocessing step. An exploration of word segmentation algorithms, however, is beyond the scope of this thesis.

2.7 Consonants

As many methods for statistically exploring vowel harmony ignore consonants, it is important to understand what role consonants play in vowel harmony in order to know what impact ignoring consonants will have on model outputs. There are several ways in which consonants interact with vowel harmony (Krämer, 2003: 22). The first, in which consonants alternate in agreement with harmonic vowels, is of little concern in these models as it represents a change to the consonants rather than to the vowels. Another case, that of consonants influencing vowel harmony through one of their place features (for example, roundness from a consonant requiring roundness in following vowels), is slightly more concerning. The final type of consonant interaction is consonant blocking of vowel harmony, in which a consonant or consonant cluster blocks vowel harmony spreading. In fact, in Finnish, velar consonants can "prevent frontness from spreading" (Hulst and Weijer, 1995: 529), as seen in Table 13. For example, the first two vowels of the words *itikka* 'mosquito' and *etikka* 'vinegar' harmonize – both are front vowels, like <i>, or transparent vowels, like <e> – but the final vowel <a> is a back vowel. The harmony is able to spread across the consonant <t>, but the frontness is blocked from spreading across the velar consonant separating <i> and <a>.

Table 13: Finnish Disharmony with <k>

Noun	
itikka	'mosquito'
etikka	'vinegar'
tiirikka	'lock pick'

(Hulst and Weijer, 1995: 530)

Another example of this appears in Tunica, a language originally spoken in Central Louisiana (Ethnologue, 2009) and for which revitalization efforts are currently underway (Foster, 2011). Tunica exhibits back and round harmony targeting only low vowels. The harmony is blocked by all consonants except for laryngeal consonants (Krämer, 2003: 23).

2.8 *Autosegmental Analysis*

Vowel harmony can be approached from the viewpoint of autosegmental phonology (Goldsmith, 1976), which has played a role in shaping NLP approaches to vowel harmony.[3] Most computational and statistical approaches to vowel harmony separate the vowels from the consonants and then compute statistics on the vowels as though they were actually adjacent to one another. This brief overview of the autosegmental analysis serves to provide a basic background of theoretically-based support for the simplifying assumptions made in modeling vowel harmony, but does not delve into the nuances of the theory. For a deeper analysis, see Clements (1977).

In the autosegmental analysis, a set of harmonizing features should be identified and placed on a separate tier. The class of vowels with the harmonizing feature is identified, as is the set of any existing opaque vowels. The harmonizing features in the system should be associated with vowels, but they must do so following the Well-formedness Condition. This condition requires that every vowel be associated with a harmonizing feature, each harmonizing feature on its tier must be associated with a vowel, and lines of association are not allowed to cross one another (Katamba, 1989: 203-212).

3 Machine Learning and Natural Language Processing for Vowel Harmony

The first question when considering using statistical machine learning and natural language processing to study vowel harmony is the question: what makes a model “good”? A successful and appropriate model for vowel harmony should meet most, if not all, of the following criteria. First

[3] Other frameworks for analysis do exist, including Optimality Theory (Prince and Smolensky 1993/2002). They are not discussed here because the statistical models in this thesis are not intended to provide a judgment regarding the relative merits of the potential theoretical explanations of vowel harmony.

and foremost, a model should have interpretable results. It should be clear to a user upon examining the results of a model what exactly it entails. This can take several forms: distilling down to a few normalized numerical results, presenting a visualization of the results, simply producing a "yes", "no", or "uncertain" result, or some combination of these. In addition to being interpretable, the models should be language agnostic and produce results that are comparable across languages and datasets. Language agnostic, in this sense, means that the model can be applied to data from various languages as long as the data is formatted correctly. Ideally, models should be flexible enough to pick out various types of vowel harmony and should be able to account for neutral vowels. However, if that is not the case, it is important that their limitations be explicitly addressed, so as not to lead to false conclusions. As the goal of this paper is to provide computational approaches to "discovering" or "diagnosing" vowel harmony from data as a first pass tool for linguists, it is not crucial that every method provide an in-depth analysis of the complete vowel harmony system in a language. Instead it should point linguists in the direction of interesting data and phenomena to explore. Since this is framed as a tool for linguistic analysis, models and data preprocessing decisions should have foundations supported by linguistic assumptions. The models need not explicitly model the desired phenomena in an identical fashion to human learners or to the linguistic processes occurring in the data, but they should provide results consistent with linguistic analysis. Finally, it is not good enough for a model to find vowel harmony where it exists – it should also fail to find vowel harmony where there is none to be found.[4]

Other preferred criteria include being unsupervised, fast, and applicable to both text data from the orthography and phonetic transcription data. The preference for unsupervised methods arises for several reasons – in order to use computational methods as a diagnostic tool for vowel harmony, it is important that a model be able to take data as its sole input, rather than requiring

[4] Ellison (1994: 2) provides a similar, shorter, set of criteria: the tool must provide a statement of generalization, which much be linguistic and "motivated by linguistic concerns", and "the analyses should be specified a priori as little as possible."

users to make predictions about the types of vowel harmony they expect to see. On the other hand, in order to use such tools for quantifying vowel harmony, it can be useful to allow users to specify what it is they are seeking. Speed of the methods is desirable for the typical reasons of user satisfaction. Applicability to both text and transcription data is desirable in terms of making the tool more widely useful, though applicability to text data will typically hinge on a close relation between the orthography and pronunciation of a language.

4 Considerations in Data and Preprocessing

In order to apply computational methods to linguistic data, it is necessary to first consider the data. Depending on the focus of the research, different types of datasets will be appropriate: for syntactic analysis it might be preferable to draw from long written or spoken texts, while for morphological analysis it may be simpler to disregard context to some extent. To study vowel harmony, one can typically be content with wordlists. When compiling wordlists to use, one must consider the possibility of word segmentation issues that can obscure vowel harmony or vowel harmony-like processes. In the case of Chamorro, a language that exhibits vowel fronting, using words segmented according to the orthography would result in missing the vowel harmony, because the triggering particles are typically written separately (Mayer et al., 2010). For this research, though, I focus only on vowel harmony constrained by word boundaries, allowing that this may result in missing other interesting phenomena on occasion. Given this concession to practicality, wordlists are an appropriate type of dataset to use. The next questions are where to acquire the wordlists and what sorts of wordlists to use.

4.1 Data sources

There are two main types of sources that should be considered, each with its own advantages and disadvantages. The first source type is wordlists comprised of phonetic transcriptions in IPA or any other internally consistent phonetic transcription system. For example the online CMU Pronunciation Dictionary (Carnegie Mellon University) for English uses ARPAbet, which maps English phonemes to strings of ASCII characters.[5] Since the study of vowel harmony concerns itself with phonology and because phonological transcriptions can provide more accuracy than orthography, phonetically transcribed texts are in many ways ideal for use as datasets. Additionally, for languages with no writing system, this may be the only type of dataset available. However, it does come with drawbacks. First, accurate transcriptions take time to produce, while texts in a language's orthography may be more readily available. Second, one has to consider the possibility of transcription error and inconsistency. Computational approaches to dialect research can face the problem of "creating so-called *Exploratorendialekte* ('explorer dialects'), i.e. 'dialects' created not by differences in pronunciation but by different people transcribing them" (Heeringa and Braun, 2003: 258). This is more likely to be a problem when dealing with differences between narrow and broad transcriptions, but could still be a problem when comparing results across languages if different datasets follow different transcription conventions. Some of these problems can be resolved by focusing only on vowels as well as careful preprocessing of data. Additionally, one needs to consider the source of the transcribed data and whether it is likely to have inflected forms, which are important as they often display vowel harmony (Mayer et al., 2010).

If one chooses to use text data in a language's orthography rather than phonetically transcribed data, one faces a different set of strengths and weaknesses. For languages with writing systems and published materials, it may be easy to find matched corpora such as Bible translations. Readily available ASCII text online may also require less preprocessing than IPA transcriptions. Naturally occurring texts and translations will also provide inflected and morphologically complex forms of words, which can exhibit vowel harmony in ways beyond those contained in uninflected forms. For example, in the case of Turkish, palatal vowel harmony is observable in the plural suffix *-lar/-ler.* In the word *ev-ler* 'houses', the front version of the suffix can be observed, paired with the

[5] For Vowel Harmony Calculator input, Harrison et al. (2004) use capitalization conventions to represent IPA vowels in a more ASCII-friendly format. The preference for ASCII symbols is due to ease of input and manipulation on a computer, rather than for human readability reasons.

front vowel <e> from the singular form *ev* 'house'; in *top* 'ball', however, the plural form is *top-lar* 'balls', with <a> matching <o> with respect to backness. This also occurs in the plural form of *adam* 'man', which is *adam-lar* 'man', though the other two examples are presented first to make it clear that it is not simply the case that the suffix uses the same vowel as the stem when in fact it harmonizes with respect to the backness feature.

Table 14: Suffixes in Turkish Harmony

Singular	***Plural***	***Gloss***
adam	adam-lar	'man'
ev	ev-ler	'house'
top	top-lar	'ball'

(example from Mayer et al., 2010)

For this reason, a list acquired from a dictionary is less than ideal, unless it is known to contain inflected forms. There is one other major requirement for using text data: the orthography must have a one-to-one or nearly one-to-one correspondence with pronunciation. Without such a correspondence, the resulting models will inaccurately represent the phonology and produce ambiguous and uninformative results. Additionally, it is useful to be able to map all vowels to the same representations (for example IPA) in order to do cross-language comparisons.

Whether one uses transcribed data or orthographic data, it is important that datasets be comparable. In a 2010 publication on vowel harmony visualization, Mayer et al. approach this problem in several ways. Their corpora consist of Bible texts in various languages, thus providing consistency in the source material as well as a dataset that contains inflected forms. They also produced random wordlists to experiment with the convergence of their method; this could also be used to ensure that results are not skewed by a single odd dataset, such as one containing a large number of loanwords.

4.2 Types vs. Tokens

After settling on a type of dataset to use, one has to decide between the use of *types* and the use of *tokens*. Given a dataset, the types can be thought of as the vocabulary of distinct words in the dataset, counting each only once. The tokens are the instances of the types. For example, suppose that the following quote is the dataset in question:

> "Far out in the uncharted backwaters of the unfashionable end of the Western Spiral arm of the Galaxy lies a small unregarded yellow sun." (Adams, 1979: 1)

The sentence contains 24 tokens, but only 19 types because there are 4 tokens of the type "the" and 3 tokens of the type "of".

Choosing between types and tokens may require some consideration of what it means to quantify vowel harmony. Using tokens would assume that what matters is the frequency of vowel harmony in natural speech or text, thus putting more emphasis on frequently occurring words. Using types would assume that frequency is more important in the lexicon as a whole, so frequent words and infrequent words will contribute equally (Baker, 2009: 5-6).

On the one hand, it seems more appropriate to use types than tokens because they better describe the overall phonology of the language, rather than focusing on only the most frequent items, which, in some languages, may be short and skew the results. The use of types also has a practical advantage, in that it helps prevent models from biasing themselves toward quirks in the data source. For example, the data used by Mayer et al. contains many frequently-occurring proper names which "in many Bible translations […] were not adapted to the phonology of the recipient language or at least not according to its common vowel patterns" (Mayer et al., 2010: 11).

On the other hand, the use of types comes with its own problems. As mentioned before, corpora with inflected forms are preferred, due to the ways that harmony can appear in affixes. Choosing to use types runs the risk of over-representing words with large numbers of inflectional forms. For example, using a type corpus with inflected forms in Spanish would result weighting the corpus toward verbs (with their many inflections) and away from nouns (which have fewer). Another downside to type corpora is that they tend to be smaller than available token corpora, which means that the models have less data to work with. Token corpora are also likely to be more accurately representative of the input that native speakers receive during language acquisition. Though this is more important for models that seek to describe cognitive processes behind vowel harmony acquisition, it is nonetheless worth considering even for models that do not claim to be related to the types of learning that occur in human language learners.

In Section 10 I present results from both type and token corpora.

4.3 Corpus Size

The larger a corpus is, the more information available to the model. Thus it is possible to be more confident that results from models run on large corpora better describe the language's phonology as a whole than results from models run on small corpora. While a large corpus is desirable, such data is not always readily available. For this reason it is important to know how small a dataset can be while still producing trustworthy results. Following methodology set out in Mayer et al. 2010, appropriate minimum corpus size can be determined empirically by choosing a gold standard result (typically from a model run on a very large dataset) for a number of languages, determining a metric by which to compare results to the gold standard, and plotting the convergence to the gold standard as models are trained on datasets of increasing size. Particularly with small dataset sizes, it is important to calculate the distance from the result to the gold standard result over a number of randomized trials of the same dataset size, in order to model typical performance. While this thesis does not contain an evaluation of the minimum corpus size that is appropriate to use, it is shown in Section 10 that corpora as small as 3126 words from harmonic languages produce results comparable to those of much larger corpora.

While this thesis provides results from models trained on entire corpora, it also presents some methods for averaging across multiple runs of a given model. As mentioned in Section 18, the large non-random corpora can be seen as representing a vocabulary (types) or the kind of input that native speakers receive (tokens). This makes them a reasonable choice for the models' datasets; all my models in this thesis are run on such datasets. However, in order to avoid the pitfalls of either dataset type, it would be appropriate to use multiple model runs on randomly chosen samples of the data.

4.4 Tier-Like Approaches

The fact that many of the models appropriate for use in studying vowel harmony work by splitting phonemes into two (or more) groups stands in favor of using a preprocessed vowel-only corpus. Since the models are likely to learn the most obvious groupings – the vowel-consonant distinction – it is preferable to remove consonants and allow models to work only on vowels. This also allows for modeling the long-distance phenomena directly, without having to take into account intervening consonants that could water down probabilities. Naturally, this choice does come at a cost – models aren't necessarily able to pick up on or provide information about consonant influence on vowel harmony or vowel harmony influence on consonants.

Separating vowels from consonants creates a tier-like approach to analyzing vowel harmony. While it doesn't perfectly match the kind of tier-based analysis used in autosegmental phonology, it approximates it. As such, it makes it possible to justify the use of this simplification – made for the sake of improved computation – in terms of linguistic theory. Additionally, it can be justified in terms of the criteria for machine learning methods stated earlier: unsupervised methods are preferred. There are well-established algorithms for automatically separating vowels from consonants, two of which are discussed in Appendix I. I choose to use vowel lists compiled in advance instead of an unsupervised method for several reasons. In order to get meaningful data on vowel harmony, it is important to know the phonological features of each vowel; an automatic vowel-identification algorithm will not extract this information. Additionally, I have presupposed either a data set containing phonetic transcriptions or a data set containing orthography with a near one-to-one mapping to the IPA. It is not unreasonable, then, to assume that I am also aware of what the mapping is so that I may accurately examine the shared phonological features of harmonic systems discovered by the vowel harmony algorithms. Nonetheless, those concerned with maintaining as unsupervised a method as possible may content themselves with the idea that automatic identification of vowels from text is possible, though unnecessary.

5 Unsupervised Methods for Quantifying and Diagnosing Vowel Harmony

Having shown that it is possible to produce useful and appropriate tiered data in a supervised or unsupervised manner, I can now examine methods for modeling vowel harmony. Ellison (1994: 2-4) describes three categories of machine learning research – connectionist, statistical, and symbolic – that can be applied to phonology. This thesis focuses on statistical learning because the phonological phenomenon of vowel harmony is well-suited to statistical machine learning and natural language processing (NLP) approaches. The probabilistic nature of such approaches enables them to explain vowel harmony systems while also handling irregularities without having to write new rules (as a symbolic approach might require). Additionally, they are better at "communicating generalizations" about the data than connectionist models tend to be (Ellison, 1994: 3) – neural networks, for example, learn weights that may not be easily interpretable, while many statistical models can be framed in terms of parameters that are easier to understand and more transparent in their workings.

In this thesis I present a new model for discovering and quantifying vowel harmony, the Mixture of Unigrams Model. In order to place it in the context of existing research, I first present the most closely related existing models for vowel harmony – Hidden Markov Models. Goldsmith and Xanthos introduce the use of Hidden Markov Models (a commonly used model in natural language processing) for vowel harmony in their 2009 paper *Learning Phonological Categories*. Baker expands on this work in *Two Statistical Approaches to Finding Vowel Harmony* (2009). Since my focus is on easily-interpretable models of whole-word harmony that can be adapted for visualizations and additional quantitative measures, I mainly discuss these models and their potential connections to other statistical methods. Other work on quantifying and learning vowel harmony patterns in Finnish has been done by Goldsmith and Riggle (to appear) in the context of information theory using unigram, bigram, and Boltzmann models. Work on visualizing pairwise harmony appears in Mayer et al. (2010). For more information on related work, see Appendix II: Other Models. In addition to presenting a new model, this thesis adds to the existing research on statistical methods for vowel harmony by introducing a new visualization tool for whole-word harmony and providing evidence of the usefulness of both the model and the visualization tool on six different languages with varying types and degrees of harmony.

6 HIDDEN MARKOV MODELS

6.1 *THE MODEL*

The first method presented in Baker's *Two Statistical Approaches to Finding Vowel Harmony* models vowel harmony using Hidden Markov Models. A Hidden Markov Model (henceforth HMM) is composed of a set of N states, additional special "start" and "end" states, a matrix A representing the transition probabilities for moving from state to state, a sequence of T observations, and a set of emission probabilities for each of the N states that express the probability of producing a given observation in a given state. The word "Hidden" in the model name refers to the fact that the states are "hidden" or unobserved. The observations are drawn from some vocabulary $V=\{v_1,v_2,\ldots,v_V\}$ (Jurafsky and Martin, 2009: 177). Having started in the "start" state, at the first time step the model moves with some probability into one of the N states and then outputs an observation based on the emission probabilities for that state. At the next time step, it repeats the process, producing a sequence of observations written as $o_1o_2\ldots o_n$ and a sequence of state variables written as $q_1q_2\ldots q_n$. The subscript denotes the time step at which the state and observation occurred. Note that it is possible to transition from a state back into that same state and even to emit the same observation multiple times in a row, depending on the transition and emission probabilities. This process continues until a transition to the end state occurs. HMMs are often drawn using a diagram as in Figure 1. Following the convention from Baker (2009), this diagram omits reference to the start and end states (which do not emit observations).[6] The circles represent states, with the directional edges representing transition probabilities.

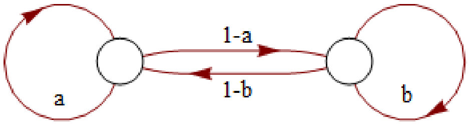


Figure 1: Hidden Markov Model

Here I present a short example of how a HMM could perform. Suppose that I have a two-state HMM where one state emits English consonants more frequently than vowels and one state that emits English vowels more frequently than consonants – call them *C* and *V*, respectively. For the sake of simplicity, I use English orthography rather than IPA. Beginning in the start state, I flip a weighted coin and move into the state *C*. At this point, I roll a weighted 26-sided die, which lands on the letter *c*. In the next timestep, I flip another weighted coin and end up moving to state *V* (there is a high probability of this happening, as English vowels and consonants tend to alternate). In this state, the model emits the vowel *a*. So far, I have the sequence of states (q_1q_2)=(CV) and the sequence of letters or emissions (o_1o_2)=(ca). This continues in this manner until I have spelled the word "cat" and reached the end state, moving through the state sequence CVC. Of course, this is just one of many words I could have produced – given different dice rolls, I could have spelled "car", "catnip" or any number of other words.

First-order Markov models make two important assumptions. The first is called the Markov Assumption, which says that the only previous state influencing the probability of ending up in a state is the state before it. Formally, this is written $P(q_i|q_1q_2...q_{i-1})=P(q_i|q_{i-1})$; that is, the probability of being in a state at time *i* given all the previous states is equal to the probability of being in a state at time *i* given the previous state. Such a model is called a first-order Markov model because it only

[6] In fact, you may note that the pairs of transition probabilities leaving a given state will sum to 1 in this diagram, leaving no probability for reaching the non-emitting end state. I choose this to maintain consistency with Baker (2009). However, should one wish to visually represent non-emitting start and end states, one could easily do so, and simply re-envision the diagrams presented here as showing the non-final transition probabilities normalized to sum to 1.

relies on one previous state; higher order models are also possible. The second assumption is Output Independence, which says that output probabilities only depend on the current state. In terms of probabilities, this is written $P(o_i|q_1...,q_i...q_T,o_1...o_i...o_T)=P(o_i|q_i)$. Both of these are simplifying assumptions that make computation tractable.

6.2 Fitting the Model Parameters

In order to fit HMMs to data, the transition and emission probabilities should first be initialized either randomly (with the caveat that none are initialized to zero so that the finite state machines are complete) or based on some prior knowledge (Baker, 2009: 11). Then these probabilities are re-estimated using the Baum-Welch algorithm (Baum et al., 1970). This is done to maximize the probability that the HMM assigns to the corpus, making it a better fit for the data. While a fully detailed explanation of the Baum-Welch algorithm is beyond the scope of this thesis,[7] it can be understood as follows: the corpus is run through the model and each state and emission event is counted, normalized, and eventually used to update the parameters of the model, then the process is repeated – each run of the algorithm can only improve or leave unchanged the probability that the HMM assigns to the corpus (Baker, 2009: 11). It is important to note that the model and algorithm are influenced by starting parameters, and may only find local maxima. For this reason, it is worth running multiple trials with different starting parameters. A method for choosing between the multiple models is mentioned in Section 6.3.

6.3 HMMs for Vowel Harmony

The use of HMMs indicates an assumption that there is some underlying and unobserved factor that influences the surface forms that are observed. In the case of vowel harmony, it means assuming that a language's vowels can be separated into classes, and each word is expected to have most of its vowels contained in just one class. HMMs are appropriate for such a task because they

[7] Rabiner and Juang 1986 as well as Rabiner 1989 provide much more extensive information on HMMs and learning HMM parameters.

can find hidden structure based on sequences of observations (in this case the vowels). One can see that they discover and represent each harmonic class as a probability distribution over vowels, skewed in favor of the vowels in that particular class.

To model vowel harmony in a language using HMMs, one can use a two state HMM. Vowel harmony commonly has two classes of vowels, so a two state model is appropriate but may not capture all aspects of the harmony systems of a language. If one intends to apply a two state HMM to a language whose harmony system is unknown (or may not even exist), it is necessary to have an understanding of how to interpret the resulting model. There are three types of results considered in Baker's paper (2009): harmonic, alternating, and a sink.

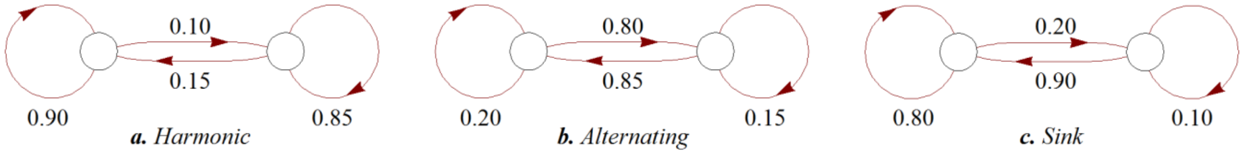


Figure 2: Three Types of HMM Results

The harmonic model (Figure 2a) has high probability of remaining in a given state, and is the only one of the three that is expected if a language exhibits vowel harmony. In a harmonic model, the probability of remaining in a given state is high and the probability of switching states is low. In terms of vowel harmony systems, this can be thought of as meaning that, given the first vowel in a word, the following vowels are likely to match that vowel with respect to the harmonizing feature.

In an alternating model (Figure 2b), there is a high probability at each timestep of transitioning out of the current state into the other state. This is the kind of model that typically appears when using HMMs to separate vowels and consonants, and would not indicate harmony – on the contrary, it would indicate a high incidence of pairwise disharmony. In the example in Section 6.1, I would expect an alternating model, since one state was likely to emit consonants and the other was likely to emit vowels and English vowels and consonants tend to alternate.

Baker found that training a model on phonetically transcribed English data resulted in a sink (Figure 2c), where one state has a high probability of remaining in the same state at each time step, while the other state has a high probability of transitioning away into the sink state where it is then likely to remain (Baker, 2009: 17). Other possible transition parameter schema should also be interpreted as non-indicative of vowel harmony.

Baker demonstrates that while a two state HMM can learn either palatal harmony or roundness harmony in Turkish (with the former proving a better fit for the corpus), a four state HMM is capable of learning both types of harmony. As mentioned in Section 6.2, the model can be initialized randomly or with prior knowledge. In his experiments, Baker found that models learned using randomized starting emission probabilities resulted in models that better matched the phenomena described by linguists than models learned using empirically determined starting emission probabilities (Baker, 2009: 20-23).

Given a number of HMMs learned with different starting parameters, the "best" model and one that should be analyzed is the one that assigns the highest probability to the data. In his conclusion, Baker suggests taking this best HMM and concluding that a language exhibits vowel harmony if both states assign no more than a 30% probability to transitioning to the other state (2009: 23), though this number seems to be somewhat arbitrary rather than empirically determined. Goldsmith and Xanthos take a similar approach to determining whether a language exhibits vowel harmony based on a two state HMM, but they present the results visually by plotting each resampling of the transition probabilities from a state to itself (2009: 27). In this way, they have a visual representation of the same data that Baker uses to make his determinations. Additionally, both methods provide a way of comparing results across languages and datasets.

7 Mixture of Unigrams

This section focuses on a related approach to statistically modeling whole-word harmony. I begin with a brief description of *n-gram models* and their applications, proceed to a discussion of the proposed model and how to fit the model, and finish by exploring the relationship between HMMs and Mixture of Unigrams models.

7.1 N-Gram Models

N-gram models (also broadly called *language models*) are probabilistic models that can be used to predict sequences of items (in NLP contexts, typically words, letters, or phonemes). Here I focus on the simplest *n*-gram models: the unigram (or 1-gram) and bigram (or 2-gram) models, with a brief discussion of the general case. For a more complete introduction to *n*-gram models and their applications, see chapter 4 of *Speech and Language Processing* (Jurafsky & Martin, 2009). In addition to predicting the next item in a sequence, *n*-gram models can be used to assign probabilities to sequences. For example, if I have an *n*-gram model for letters in English text and an *n*-gram model for letters in Spanish text, I can assign probabilities under each model to a string of letters and make a prediction based on those probabilities as to whether the word is most likely Spanish or English.

When I talk of making predictions about sequences in the context of natural language processing, there are a number of example applications. To do automatic speech recognition or optical character recognition, for example, it is appropriate to have both a language model and an acoustic or image recognition model, respectively. This allows the model to make predictions about how likely different sequences of letters are based both on their auditory (or visual) representation as well as a background understanding of the language being used. For example in the optical character recognition case, if one were examining a smudged English text and were confident that the first letter in a word were “w” and thought based on the image recognition that the next letter was probably “h” or “b” (which look fairly similar), one could use a language model to find out that the next letter is much more likely to be “h”. This is due to the fact that the bigram (pair of letters) “wh” is much more common in English than the bigram “wb”.

With this brief introduction, I now give a slightly more formal definition of an *n*-gram model. I begin with a sequence of random variables $(\mathrm{X}_1, \dots, \mathrm{X}_\mathrm{n})$, which will sometimes be written X_1^n for convenience. Using the definition of conditional probability, I assign probability to the sequence by doing: $P\big((\mathrm{X}_1, \dots, \mathrm{X}_\mathrm{n})\big) = P(\mathrm{X}_1)P(\mathrm{X}_2|\mathrm{X}_1)P(\mathrm{X}_3|\mathrm{X}_1^2) \dots P(\mathrm{X}_\mathrm{n}|\mathrm{X}_1^{\mathrm{n}-1}) = \prod_{\mathrm{k}=1}^{\mathrm{n}} P(\mathrm{X}_\mathrm{k}|\mathrm{X}_1^{\mathrm{k}-1})$. Unfortunately, this alone isn't enough to be useful. One has to consider the fact that these models will eventually be trained using corpus data, which is an incomplete representation of human language. What this means in practice is that computing the probabilities as they stand based on a corpus will be inaccurate for long sequences – as *n* increases, the number of possible sequences also increases, and it is unrealistic to expect to see all of them in a corpus. This combination of problems will result in improperly small (or even zero) probabilities assigned to long sequences (Jurafsky & Martin, 2009: 87-89). Instead, it is possible to make a familiar simplifying assumption to approximate the probability: the Markov assumption. In the bigram case, I can say that $P(X_n|X_1^{n-1}) \approx P(X_n|X_{n-1})$, making this equivalent to a first-order Markov model. In the general case, an *n*-gram model is equivalent to an *n-1* order Markov model.

Learning an *n*-gram model is typically done using Maximum Likelihood Estimation on a corpus. Using a bigram model as an example, the probability $P(X_n|X_{n-1})$ is estimated by counting all instances of the bigram $X_{n-1}X_n$ and normalizing by the count of all bigrams whose first element is X_{n-1}, which is the number of times X_{n-1} appears in the corpus. For higher order *n*-gram models, or for cases where the corpus may be incomplete, it is useful to do smoothing. This distributes small amounts of probability mass to unseen items or to items with zero probability mass in order to be sure that no sequence of items is assigned zero probability. This avoids the potential problem of assigning zero probability to perfectly valid but as yet unobserved sequences. However, since vowel inventories will be known in advance, the models discussed in this thesis will not require smoothing.

7.2 *Example of a Unigram Model*

Imagine having two languages – L1 and L2 – that share a common set of written characters, the two-letter set {A, Z}. First consider a unigram example where the probabilities are estimated from large corpora of each language: choosing a letter at random from a text in L1 leaves you with a 50-50 chance of choosing A or Z. In texts from L2, you have a 70% chance of choosing A and a 30% chance of choosing Z. Given the word AAAZ, is it more likely that this word is from L1 or L2? Under the unigram model, P(AAAZ)=P(A)·P(A)·P(A)·P(Z). Under the L1 model, the product is $P(\text{AAAZ}|\text{L1}) = 0.5 \cdot 0.5 \cdot 0.5 \cdot 0.5 = 0.0625$. Under the L2 model, the product is $P(\text{AAAZ}|\text{L2}) = 0.7 \cdot 0.7 \cdot 0.7 \cdot 0.3 = 0.1029$. Thus AAAZ is more likely to be a word from the language L2, since 0.1029>0.0625. It would also be possible to base conclusions about the same word on bigram or *n*-gram models.

7.3 *Mixture Model*

Instead of assuming that the data can be well-represented by one unigram model, it might be preferable to think of the data as coming from two unigram models. This intuition – based on both my own investigations and the harmonic HMMs from Baker (2009) – comes from the observation that many vowel harmony systems exhibit a two-way distinction (e.g. front words vs. back words). If a language has two vowel harmony classes, then it would be expected that most words would fall into either one class or the other. Thus it would be appropriate to have one unigram model to model probabilities of the vowels in words of one class, and another to model the probabilities of the vowels in words of the other class. The resulting graphical model is called a *Mixture of Unigrams* model. For information about how such a model compares to other graphical models with NLP applications, see Blei, Ng, and Jordan (2003). The Mixture of Unigrams model can compare directly to the HMM model if thought of as a HMM with zero probability of transition between the two classes. The model can be explained using a generative story, which is the simplified way to *imagine* data was created (clearly, this is not the true process for word creation but rather a framework for modeling a much more complex phenomenon).

To provide a short example of the generative story for the Mixture of Unigrams model, I present an example from a hypothetical four vowel system that exhibits palatal harmony. The vowels /i/ and /e/ form the class of front vowels, while /u/ and /o/ form the back class of vowels. I imagine that I am creating the vocabulary of a constructed language with this harmony system based on a Mixture of Unigrams model – for the sake of this example I'll assume that all words have exactly 3 vowels in them. Each time that I want to create a new word, I first flip a weighted coin to decide whether it will be a front word or a back word. There is a 60% chance that I will choose the front class, and a 40% chance that I'll choose the back class (Figure 3).

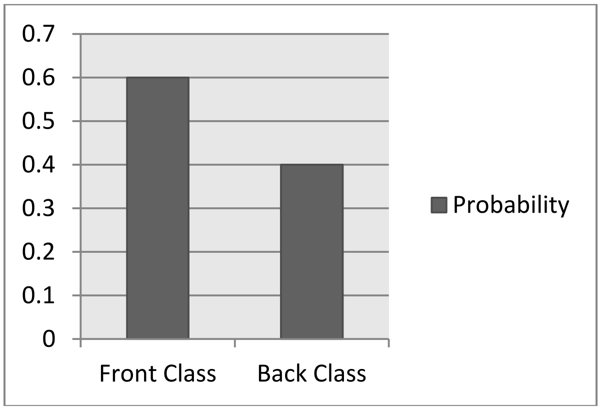


Figure 3: Mixture of Unigrams Class Probabilities

Each of these classes can be represented by a probability distribution over the set of vowels (Figure 4). The front class assigns higher probability to the front vowels, while the back class assigns higher probability to the back vowels. It is worth noting, though, that none of the vowels are assigned zero probability in either class – this allows for the possibility of producing disharmonic words (as harmonic languages will often do).

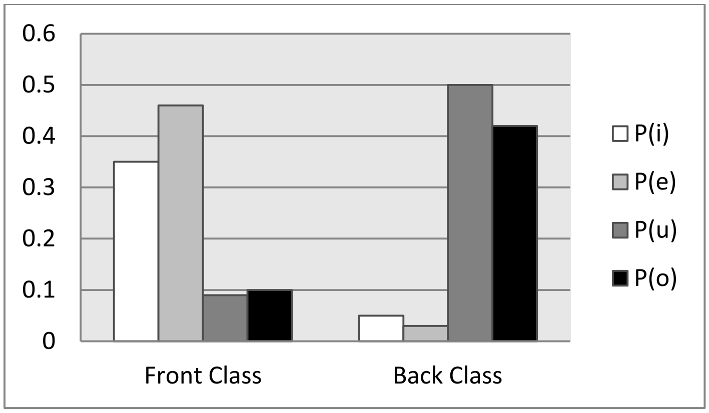


Figure 4: Mixture of Unigrams Vowel Probabilities by Class

Now I decide that it's time to create a new word in my constructed language. Having flipped the coin, suppose that it lands on the front class. For each of the three vowels in the word, a die weighted according to the probability mass function for the front class (Figure 4) is rolled. If it lands on, for example, the sequence /iee/, then those will be the vowels in the new word I am creating. Note that this probability mass function is more likely to produce a harmonic set of vowels like /iee/ from the front class than a disharmonic one like /ieo/, based on the probability mass function (though /ieo/ is not impossible). Section 7.4 explores the relationship between this generative model – which is certainly not representative of the true story behind vowel harmony systems in naturally-occurring language – and how to fit the model using actual data from corpora. For a more formal approach to both the model's generative story and how to fit the model, see Appendix III.

Now it is necessary to consider how and why this model is appropriate for vowel harmony tasks in real languages and not just this hypothetical one. For example, to describe Finnish, it might be useful to use a mixture of two unigram models – one that assigns high probability to front vowels and thus generates the front words, and one which assigns high probability to back vowels and thus generates back words. Since these unigram models are probability mass functions over the set of vowels in the language, the model still allows for some disharmony by assigning small but non-zero probabilities to disharmonic vowels. Given a non-harmonic language, one would not expect the same split into two harmonic classes, so it might be reasonable to guess that each unigram model would more or less match the overall probability mass function of the vowels in the language and be used with near-equal frequency. Alternatively, there might be a split where one unigram model matches the overall probabilities of the language's vowels and is used much more frequently than the other model (which could pick up on some subset of words with a different set of vowel probabilities – e.g. consistent borrowings from some other language). Either of these options would indicate a non-harmonic language. On the other hand, a clear split into two coherent classes of vowels should indicate a whole-word harmony system. Such a clear split would have a low cosine similarity between the probability vectors representing the classes (simply put, vowels with high probability in the first class would have low probability in the second class – and vice versa). Additionally, the set of high probability vowels in each class should be grouped according to some coherent feature (like backness, height, or roundness).

7.4 Fitting the Model, Convergence, and Implementation

While the Mixture of Unigrams model was described using the generative story, the true interest is not in *generating* data but in fitting model parameters given existing corpora. The technique used here is a Markov Chain Monte Carlo method called (collapsed) *Gibbs Sampling*. While Section 7.4 and Appendix III contain brief overviews of the method for fitting the model, interested readers looking for a longer introduction to Gibbs sampling may wish to consult Resnik and Hardisty's technical report *Gibbs Sampling for the Uninitiated* (2010).

Informally, Gibbs sampling for Mixture of Unigrams models works as follows: all words in the dataset are randomly assigned to one of the classes. Counts of the vowels in each class are then incremented to reflect the current probability distributions over vowels for each class. For a set number of iterations, the sampling repeats itself by cycling through each word in the corpus. For each word in the corpus, its vowel counts are subtracted from the total counts, then its probability under each class is calculated, and the class label for the word is redrawn based on those probabilities. The counts are re-incremented accordingly (so that the new probability distributions take into account the word's new class), and then the process repeats with the next word.

Gibbs sampling is sensitive to initial conditions and sometimes finds local maxima rather than global maxima – this means that, while it may find a good set of probability distributions, it may not find the set of probability distributions that best describes the data. It is therefore advisable to run Gibbs sampling multiple times on a corpus. From there, one can either choose the model that assigns the highest probability to the corpus or average across models. It is worth noting that, due to randomness, the class numberings may be different across multiple runs. For example, running Gibbs sampling for a Mixture of Unigrams model might call front vowels class 0 in one run and class 1 in the next. While this could be a problem if there were a large number of classes, it is simple enough to find the pairs of similar classes (using a measure like cosine similarity).

In addition to running Gibbs sampling multiple times on a corpus in order to avoid local minima, averaging over multiple runs on randomly chosen subsets of the corpus can help avoid overfitting to quirks of the dataset (such as loanwords). Section 10 presents results for single runs, but future work on this topic could include an exploration of the impact of single run as compared to aggregated run results, particularly for smaller datasets.

The code used for the Mixture of Unigrams models in this thesis is an expansion of earlier code by Wallach, Knowles, and Dredze (2011) from our work on extensions to topic models.[8]

7.5 Relation to HMMs

A Mixture of Unigrams model can be viewed as a special case of a two-state HMM with zero probability of transitioning from one state to the other state. This has several consequences for the use of such models. Both HMMs and Mixture of Unigrams models are well-suited to the discovery

[8] Latent Dirichlet Allocation and related extensions to topic models can discover semantic topics in corpora of documents based on document-word co-occurrences. It was through my work with Hanna Wallach and Mark Dredze that I became interested in exploring similar (but simpler) models for studying vowel harmony through word-vowel co-occurrences.

of whole-word harmony, and can thus face muddled results in situations where harmony is blocked within the word. However, given their statistical nature, they can account for some disharmony, so it is still appropriate to use them to measure how harmonic a language is. The fact that the Mixture of Unigrams model is a special case of the HMM model begs the question of the strengths and weaknesses of the respective models. The Mixture of Unigrams model benefits from its simplicity. To do cross-language comparison using Mixture of Unigrams, there are fewer variables that must be accounted for. Therefore one could propose a measure for harmony based on a simple distance metric between the two distributions. Such a measure in HMMs is more complicated, as one must also take into account the transition probabilities. Additionally, HMMs make an assumption about the directionality of harmony; according to the Markov assumption each state (or in this context, harmonic class) depends on the previous state. Mixture of Unigrams models need not make such an assumption, rendering them less closely tied to one particular type of harmony system. Additionally, the fact that it is possible to learn, visualize, and model harmony systems using a Mixture of Unigrams model (which makes a stricter whole-word harmony assumption than HMMs) helps to satisfy the goal of having a simple model.

8 Non-Generative Models

The following model for analyzing vowel harmony differs from the previously mentioned models in that it is not generative. That is, rather than creating models which could be used to assign probabilities to more data in the given language, the next model only provide statistical measurements of the dataset. To examine more data from the language the computations would need to be repeated, while the generative models can assign probabilities to new data based on the parameters they have fitted.

9 Vowel Harmony Calculator

The Vowel Harmony Calculator (VHC) created by K. David Harrison, Emily Thomforde, and Michael O'Keefe is an online tool for studying vowel harmony at the whole-word level. While the

VHC also offers a Conditioned Harmony calculator in the testing stages, I focus on the VHC Unconditioned harmony tool only. Based on a corpus of ASCII text, user-provided description of the harmony system (e.g. list of Finnish front vowels, neutral vowels, and back vowels), and user-specified options on long vowels and diphthongs, the VHC can be used to produce a quantification of a language's vowel harmony. It can handle languages with and without transparent vowels. In calculating statistics on whole-word harmony, it ignores monosyllabic words (which are by definition harmonic).

What the VHC seeks to provide is a measurement of how much a language harmonizes. This is not as simple as just counting the number of harmonic words (that is, words containing vowels from only one harmonic class). Since some vowels may be more common than others in a given language, it is possible to have "class skewing", which means that one harmonic class is more common than another. In order to resolve this, the VHC calculates a "harmony threshold", or the percentage of words that one would expect to be harmonic based on chance. This threshold considers both the overall vowel distribution of the corpus (and potential class skewing) as well as the average syllable count of words (ignoring monosyllables).

The harmony threshold is a baseline against which to compare the actual percentage of harmonic words, which is calculated directly from the data. The "harmony index" is the actual percentage of harmonic words in the corpus minus the harmony threshold. A large harmony index indicates a highly harmonic language, while a small harmony index indicates little to no harmony.

In addition to producing a harmony index, the VHC provides users with quite a bit more data, including measures of harmony on initial two syllables, a log of the disharmonic words, and vowel frequency information, among other things.

10 Results

In this section I present results from both the VHC and Mixture of Unigrams. The data was not run through HMMs, so I cannot provide direct comparison of that model, but I suspect that my results would match up well with results of the methods presented by Baker (2009) were I to run HMM experiments on these datasets, as the HMMs find comparable distributions over vowels for Finnish and Turkish in Baker's experiments.

10.1 Data

In order to ensure compatibility with both the VHC and Mixture of Unigrams models, the datasets chosen are ones available online from the sample corpora and additional results on the VHC website (Harrison et al., 2004). The corpora were chosen to cover both palatal and height harmony, as well as non-harmonic languages. The specific corpora[9] chosen for each language were chosen based on their similar sizes. The token and type counts (Table 15) are based on the corpora with monosyllabic words removed, as monosyllabic words are ignored by the VHC calculations. In order to better compare the results from the two models, the Mixture of Unigrams models were run on these corpora with monosyllabic words removed even though that is not a requirement of the model (a comparison of results with and without monosyllabic words is presented in Section 10.3).

Table 15: Dataset Sizes

Language	Harmonic	Number of Tokens	Number of Types
Finnish	Yes	17726	4779
Swahili	Yes	15018	4246
Turkish	Yes	18641	18543
Tuvan	Yes	8135	3126
Japanese	No	10967	3786
Indonesian	No	10934	1815

Before moving on to the results, there are a few brief comments to be made on the corpora used. Both the Indonesian and Finnish corpora are Bible text corpora (more specifically the gospels). The Japanese corpus is comprised of song lyrics, the Swahili text is verbs only, and the source texts for the Tuvan and Turkish corpora are not specified. While it would be ideal to have better matched corpora, these corpora were chosen based on their size similarity and relation to the VHC.

[9] The corpora are listed on the webpage under the following names: finnish-gospels [txt], Swahili-verb [lex], turkish [lex], tuvan-mark [txt], Japanese-pop [txt], and Indonesian-gspls [txt].

10.2 VHC Results

The results from the VHC show, as expected, a high harmony index for the harmonic languages and a low harmony index for the languages not expected to exhibit harmony (Table 16). The only parameters set for these runs were the types of harmony to test for (diphthongs were not reduced, and no distinction was made between long and short vowels).[10] For height harmony, the vowels are listed in the format high/neutral/low, and for backness harmony they are listed as front/neutral/back. The corpora for the languages known to be harmonic were run with the parameters set to their known harmony systems. As Japanese and Indonesian are not expected to have any harmony, they were run with the parameter set to the Swahili harmony system, since all three languages share very similar vowel inventories.

Table 16: Vowel Harmony Calculator Results

Language	Harmony Tested	Harmony Index (Tokens)	Harmony Index (Types)
Swahili	Height: iu/a/oe	20.81%	23.29%
Finnish	Backness: äöy/ie/aou	29.62%	36.24%
Turkish	Backness: ieüö/-/ɨauo	30.92%	30.89%
Tuvan[11]	Backness: ieüö/-/ɨauo	56.62%	59.39%
Japanese	Height: iu/a/oe	5.92%	5.67%
Indonesian	Height: iu/a/oe	2.68%	2.18%

It is worth considering the harmony index results for token corpora as opposed to type corpora. In all cases except Turkish, the harmony index for languages with vowel harmony is greater in type corpora than token corpora. It is worth noting that the sizes of the type and token corpora are much closer for Turkish than for the other languages. Additionally, one can note that non-harmonic languages seem to have lower harmony indices in type corpora than token corpora. In order to get a better grasp of the difference between the kinds of corpora (or the lack thereof), it would be prudent to run randomized, size-matched tests of both type and token corpora in order to see whether there truly is a difference in the results or whether these differences can be accounted for by random chance error. All in all, however, the results are very close between the two, so it seems

[10] In fact, for all models run, diphthongs and long vowels are not treated any differently from other vowels.

[11] Note: here I use the Turkish orthography for Tuvan to maintain consistency due to their similar vowel inventories, rather than switching to IPA or a Cyrillic alphabet.

safe to use either types or tokens. This is reassuring, as it minimizes the concern about theoretical implications discussed in Section 4.2.

10.3 Mixture of Unigrams Results

In this section, I introduce results from the Mixture of Unigrams model. Visualizations are presented in Section 11. For each language, four versions of the model were run:[12] token corpus with monosyllabic words excluded, type corpus with monosyllabic words excluded, token corpus with monosyllabic words included, and type corpus with monosyllabic words included. This allows for a comparison that is informative with regard to the questions raised about the difference between type and token corpora. Additionally, it helps to answer the question as to whether having monosyllabic words in a corpus is likely to skew results.

First, to familiarize the reader with the output, I present output from the Mixture of Unigrams model run on the Finnish token corpus with monosyllabic words removed (Table 17).[13] For more examples of output, see Appendix IV.

Table 17: Finnish Vowel Probabilities by Class

Class 0		***Class 1***	
ä	0.36	a	0.32
e	0.28	i	0.22
i	0.26	e	0.16
y	0.08	u	0.15
ö	0.03	o	0.15
a	≈0.0	ä	≈0.0
u	≈0.0	y	≈0.0
o	≈0.0	ö	≈0.0

Class 0 assigns higher probability to front vowels and nearly zero probability to back vowels, while Class 1 does the opposite. Both classes assign a fair amount of probability mass to both <i> and <e>, the neutral vowels; this is to be expected, since neutral vowels appear in both front words and back words. The one probability that stands out as potentially concerning is the low probability

[12] Each model was run with 5000 iterations of Gibbs sampling, though convergence appears to have been reached in well fewer than 500 iterations for all harmonic language corpora. Though it is also possible to optimize hyperparameters, this was not done for any of the models.

[13] Values presented here are rounded to two places after the decimal, and as such may not sum to 1.

mass assigned to the front vowel <ö> in Class 0. As it turns out, this is related to the low overall number of instances of that vowel in the corpus. A discussion of how to resolve this apparent problem can be found in Section 11.

Within a given language, it is possible to compare probability mass functions learned by variations of the model or dataset. This is easily done by computing cosine similarities. The cosine similarity between two vectors is computed using the formula $Similarity(u,v) = \frac{u \cdot v}{\|u\|\|v\|}$, which calculates the cosine of the angle between the two vectors. A result of 0 indicates orthogonal vectors, while a result of 1 indicates that the angle between the vectors is 0 meaning that they are similar.

The comparison of Mixture of Unigrams models is slightly more complicated. Let A_i indicate the results from one model run and B_i indicate the results of the other, where i is the index of the class. Then the similarity between the two learned models can be computed as $Maximum\ Cosine\ Similarity = \max(\frac{Similarity(A_0,B_0)+Similarity(A_1,B_1)}{2}, \frac{Similarity(A_0,B_1)+Similarity(A_1,B_0)}{2})$ and should be distinguished from just cosine similarity alone. Choosing the maximum means that matching classes are compared.

The use of cosine similarity (and a variation on cosine similarity) is appropriate as a basic measure of how harmonic a model is, but it is inappropriate as the only tool for such a judgment. Probability distributions over vowels for harmonic languages are expected to be fairly orthogonal, since each probability distribution should assign high probability to one class of vowels while assigning low probability to the other class. Thus cosine similarity will generally produce low values for harmonic languages. Of course, there are drawbacks, as we could find low cosine similarity between probability distributions that are not actually harmonic – that is, that do not divide the vowels into coherent classes based on some feature. Thus it is important to also examine the distributions for coherent vowel classes. However, this still serves as a simple and useful first test. It also has a feature that makes it preferable over other measures like Bhattacharyya distance;

its range is [0,1] rather than infinite. Since it is mainly used here for checking the differences between datasets and results on those datasets, as well as plotting a comparison against another method of quantifying vowel harmony, it is convenient to bounded values to compare.

Table 18: Mixture of Unigrams Maximum Cosine Similarity Results

	Tokens vs. Types (Without Monosyllables)	***Tokens vs. Types (With Monosyllables)***	***Monosyllables vs. No Mono. (Tokens)***	***Monosyllables vs. No Mono. (Types)***
Finnish	0.99512	0.99604	0.99583	0.99897
Tuvan	0.99493	0.99346	0.99432	0.99797
Turkish	0.99985	0.99999	0.99322	0.99252
Swahili	0.99933	0.99823	0.99676	0.99878
Japanese	0.99741	0.94707	0.91973	0.99404
Indonesian	0.99697	0.98479	0.98753	0.99354

Table 18 displays the maximum cosine similarity for various runs of the Mixture of Unigrams model. As it turns out, the model learns highly similar probability mass functions regardless of the choices made with respect to monosyllables or tokens and types. The only languages for which the similarity ever falls below .99 are Japanese and Indonesian, the non-harmonic languages, which is of little concern because their probability distributions are not likely to resemble harmonic classes. While it may seem somewhat surprising that the maximum cosine similarity is so high, an examination of the cosine similarity of the overall distributions of vowels in the corpus shows that those also vary minimally across token/type and monosyllable distinctions. Each column in Table 19 shows the cosine similarity of unigram models on two versions of each dataset. All values are above 0.98, meaning that the overall vowel probability distributions vary minimally across the different versions of the datasets.

Table 19: Cosine Similarity for Overall Vowel Probability

	Tokens vs. Types (Without Monosyllables)	*Tokens vs. Types (With Monosyllables)*	*Monosyllables vs. No Mono. (Tokens)*	*Monosyllables vs. No Mono. (Types)*
Finnish	0.99369	0.99580	0.99513	0.99857
Tuvan	0.99016	0.99248	0.99322	0.99865
Turkish	1.0	1.0	0.99859	0.99861
Swahili	0.99952	0.99931	0.99761	0.99865
Japanese	0.99849	0.98712	0.99007	0.99981
Indonesian	0.99843	0.99971	0.99045	0.99714

What this means, is that, while the token/type distinction may have theoretical implications when considering how to measure vowel harmony, its practical implications are likely minimal. Additionally, it should not be problematic to compare methods regardless of their use of monosyllabic words, though for the sake of consistency, Section 10.4 compares VHC results to Mixture of Unigrams results run without monosyllabic words.

Though I have focused on the probability mass function produced, due to its usefulness in both visualizations (Section 11) and quantifying harmony, the Mixture of Unigrams model produces other output that could be leveraged for the study of vowel harmony. First, it would be possible to produce and consider the probability distributions over harmonic classes – this could be useful in the event that one class is very strongly favored over the other, which might indicate a non-harmonic system (similar to the sink in HMMs). The other potential tool is the probability that model assigns to the corpus as a whole (typically implemented as a log probability, due to underflow issues). This is calculated at each iteration of Gibbs sampling, and is expected to converge. A brief examination of this has shown that the harmonic languages converge quite quickly, but at least one non-harmonic language (Japanese, types, no monosyllabic words) took over 1000 iterations to converge. Thus it might be possible to explore the use of time-to-iteration as a tool for diagnosing vowel harmony.

10.4 Comparison of Results

A Mixture of Unigrams model for a language with vowel harmony should have low cosine similarity between the two classes of vowels. This is an imperfect measurement of whether the probabilities learned do represent a harmonic system or not, as it would also be possible to have low cosine similarity between two classes of vowels without the groupings of high and low probability vowels having a shared feature such as backness or height. As such, it should not be depended on as the sole measure of whether a Mixture of Unigrams model fit to data has learned a harmonic or non-harmonic system. However, it serves as a convenient and simple way of directly comparing Mixture of Unigrams results to VHC results, showing that the results are well-correlated. In Figure 5 I plot the VHC Harmony index on the x-axis and the Mixture of Unigrams results, calculated as $1 - Similarity(Class\ 0, Class\ 1)$, on the y-axis.

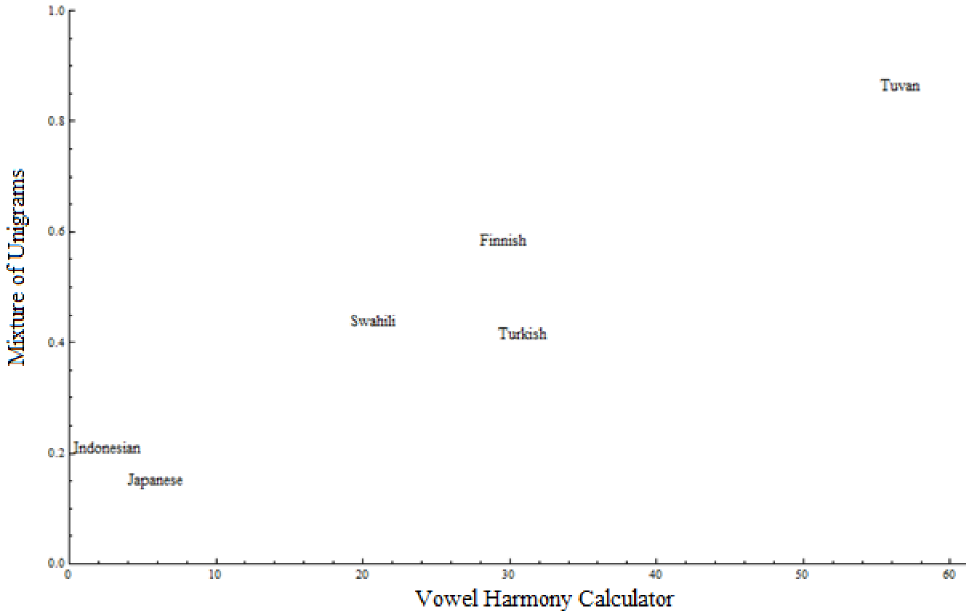


Figure 5: Vowel Harmony Calculator and Mixture of Unigrams

11 VISUALIZATIONS

One of the strengths of statistical models is their ability to quantify phenomena like vowel harmony. These results can then be presented in a number of ways. Previous work on HMMs for vowel harmony presents results in HMM diagrams and tables of values. While this provides quite a bit of useful information about the vowel harmony system, it does not allow the user to make initial judgments at a glance. Goldsmith and Xanthos (2009) provide a visualization tool for determining whether the output of an HMM is harmonic or alternating. However, the numerical data can be used to create more extensive visualizations for vowel harmony. In Section 11.1, I introduce a method for visualizing not only whether or not a language is harmonic but also, if it is, what type of harmony it exhibits.

A visualization tool provides a quick way to gain an understanding of the vowel harmony system of a language. Mayer et al. (2010) present a matrix-based approach to visualizing pairwise vowel harmony and other related phenomena. Their visualization tool provides a convenient way of seeing clusters of vowels (what I have termed vowel classes), but it does not provide immediate clarity as to what features unite the clusters of vowels, nor does it provide information about whole-word harmony. Goldsmith and Xanthos (2009) provide a visualization tool for HMMs that plots the transition probabilities from a state to itself at each step in the learning of the model. Each axis represents one state. Starting, typically, from somewhere near the point (0.5, 0.5), they plot the transition probabilities each time the model parameters are re-estimated. At the end, models with high *x* and *y* coordinates (in the upper right quadrant) are considered harmonic since they are likely to remain in the same state and models with low *x* and *y* coordinates (in the lower left quadrant) are considered alternating. However, this does not provide the user with a visualization of the type of harmony occurring, only whether or not the language is harmonic.

The visualization tool I present provides the user with visual feedback about both the type and extent of whole-word harmony in a language. When confronted with a language whose harmony system (or existence thereof) is unknown, a linguist could use this tool to rapidly determine whether or not the language appears to have harmony, and if so, how to set parameters for a more supervised quantification tool like the VHC.

11.1 HMMs and Mixture of Unigrams

The visualization method presented in this section[14] is appropriate for both HMMs and Mixture of Unigrams models, as they both produce compatible outputs. I first present example results from the token gospel corpus of Finnish (with monosyllables removed). As mentioned earlier, Finnish has an eight vowel system with palatal harmony (Table 20).

Table 20: Finnish Vowels

Front		***Back***		
Unrounded	Rounded	Unrounded	Rounded	
i	y		u	***High***
e	ö		o	***Mid***
ä		a		***Low***

The vowels can be displayed in a grid as follows, with the X-marked squares representing sounds not in the Finnish vowel inventory:

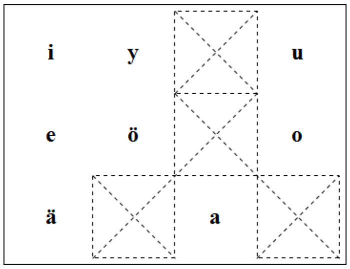


Figure 6: Finnish Vowel Template

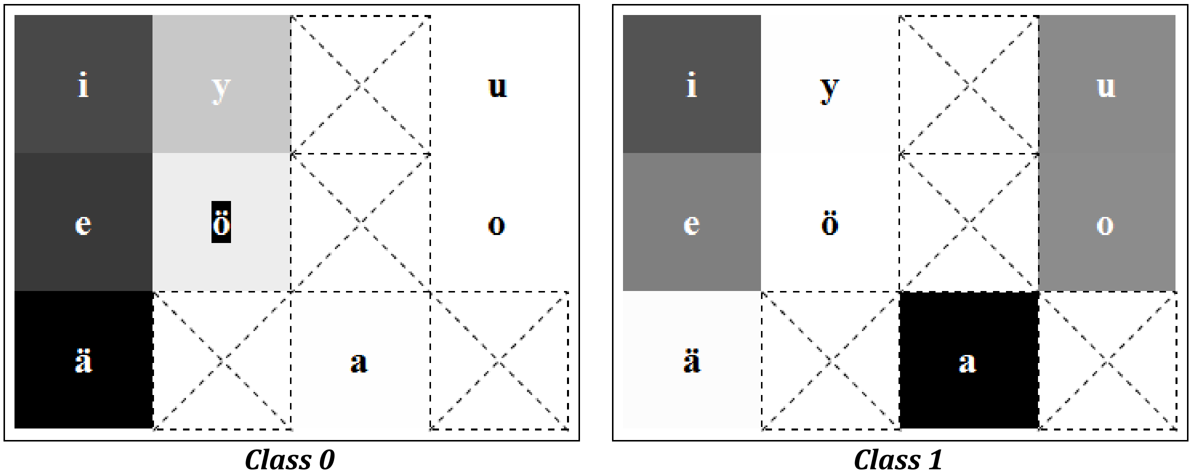
This grid will be the template for the creation of a heat map based on the model output. For each language's vowel inventory, a similar grid is created. I follow linguistic convention in representing the vowel space by placing front vowels to the left, back vowels to the right, low vowels on the bottom, high vowels on the top, and, when applicable, rounded vowels to the right of their unrounded counterparts. The model output is shown in Table 21, with the class probabilities over vowels.

[14] The graphics in this thesis were produced using Wolfram Research's *Mathematica 8* (2010).

Table 21: Finnish Vowel Probabilities by Class

Class 0		Class 1	
ä	0.36	a	0.32
e	0.28	i	0.22
i	0.26	e	0.16
y	0.08	u	0.15
ö	0.03	o	0.15
a	≈0.0	ä	≈0.0
u	≈0.0	Y	≈0.0
o	≈0.0	ö	≈0.0

These same probabilities can be displayed in a heat map – the darker the block, the higher the probability. All vowel probabilities are divided by the largest probability, which I call $PMax$, then the opacity of the squares is determined by $100 * P(vowel)/PMax$. This means that the vowel with highest probability is always shown with 100% opacity.



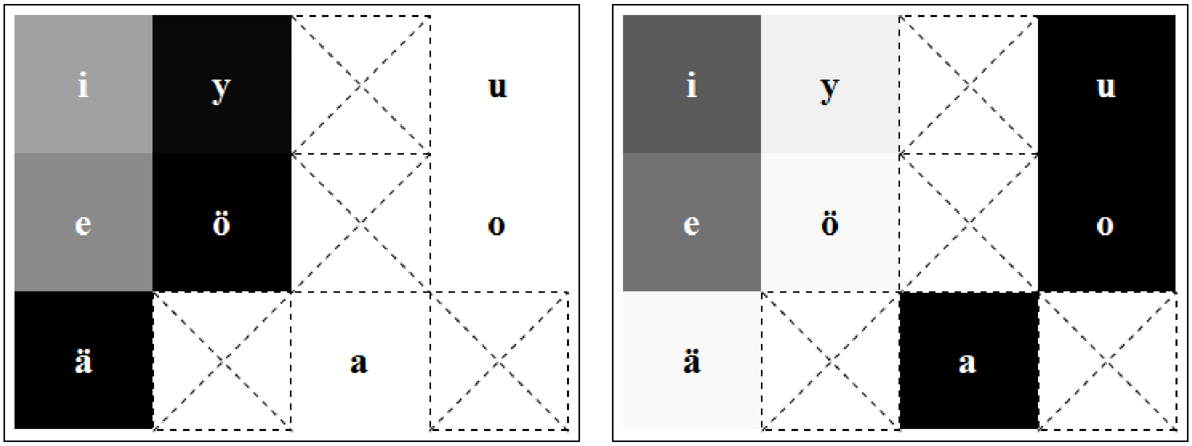
Class 0

Class 1

Figure 7: Initial Visualization for Finnish Vowel Harmony

In Figure 7, I present a simple visual representation of the class probabilities for each vowel. One can get an inkling that the vowels seem to be split up into two groups (front and back), with some overlap (the transparent vowels /i/ and /e/). The image isn't all that compelling, though, because of the visibility of the transparent vowels. Imagining that one knew nothing of the language's vowel harmony, this image wouldn't necessarily inspire all that much confidence in the notion that the language is definitely harmonic.

However, some minor changes can drastically improve the quality of the visualization's ability to display the harmonic system. Notice that the block for <ö> (highlighted in the *Class 0* diagram, Figure 7) seems to be particularly pale in both classes. This is because that vowel is less common in the corpus as a whole. In order to make sure that the visualization is not unduly influenced by the corpus-wide probabilities of each vowel, I implement a fairly simple fix. First compute the unigram model probability mass function over vowels for the entire corpus. Then divide each probability in the Mixture of Unigrams model by the overall unigram model probability for that vowel. These new numbers are now treated in the same way as explained earlier (such that the new highest ranking vowel still has 100% opacity). This has a strong positive impact on the visualization results (Figure 8).



Class 0 (Front) ***Class 1(Back)***

Figure 8: Normalized Visualization for Finnish Vowel Harmony

Now the split between front and back vowels is much clearer, while still demonstrating the role of the neutral vowels. When using this visualization tool, harmony is apparent through stripes of high opacity contrasted with stripes of low opacity. The high opacity stripes of one class should be the low opacity stripes of the other, and vice versa. Thus, palatal harmony is indicated by strongly visible vertical stripes and height harmony is indicated by horizontal stripes.[15]

Additionally, it is possible to produce a visualization that indicates whether a language has transparent neutral vowels in its harmony system. Let v_i represent the probability mass assigned to a given vowel v under the probability mass function from class i. For each vowel, the number

[15] Though I have no examples of it here, it is not unreasonable to predict that it may be possible to see labial (roundness) harmony, based on pairs of stripes in the round/not-round columns.

$n_v = \frac{v_0+v_1}{\max(v_0,v_1)} - 1$ is computed. This will always fall between 0 and 1. A similar visualization can be created where the percent opacity is equal to $100 * n_v$ for each vowel. Dark vowels are those that are have similar probability mass across both classes – likely candidates to be transparent vowels. The Finnish vowels <i> and <e> show up clearly in the visualization (Figure 9).

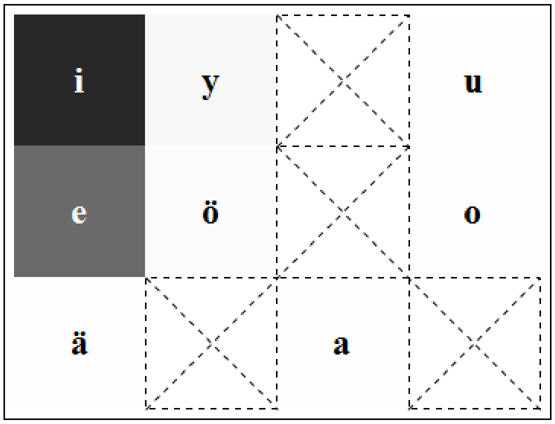


Figure 9: Finnish Neutral Vowels

Swahili, a five-vowel language, does exhibit height harmony and this shows up well in the visualization, particularly once it becomes clear that the low vowel <a> is transparent (Figure 10).

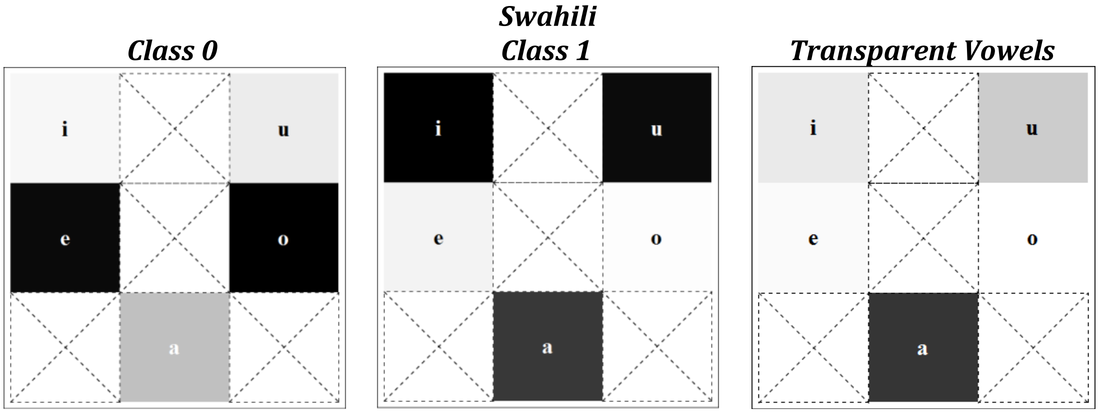


Figure 10: Visualization for Swahili Vowel Harmony

Having seen what this does for two harmonic languages, it is also important to check that both the model and the visualization tool don't find harmony where it ought not to be found. To this end, I use the Japanese pop lyrics dataset. Japanese, like Swahili, has a vowel inventory with five vowels.

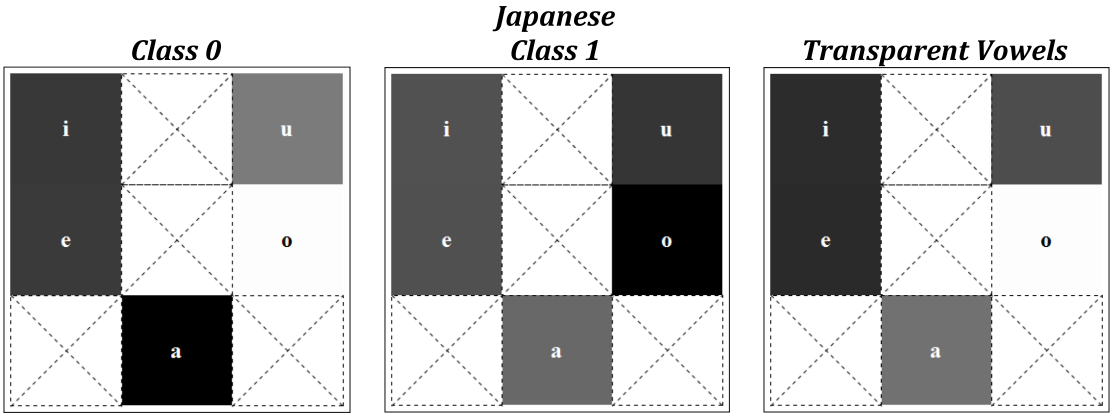


Figure 11: Visualization for Japanese

There don't seem to be clear classes for Japanese in the same way that there were in Finnish, which is to be expected because the language is known not to be harmonic (Figure 11). In fact, the visualization for transparent vowels marks all vowels but <o> as candidates for transparency. This indicates that there is strong overlap between the two classes, showing that the language does not appear to be harmonic.

Even for languages that are harmonic, there are many differences in the harmony systems. It is important to see that the transparent vowel visualization also fails to invent spurious transparent vowels. This is clearly the case in Tuvan, which has no neutral vowels (Figure 12).

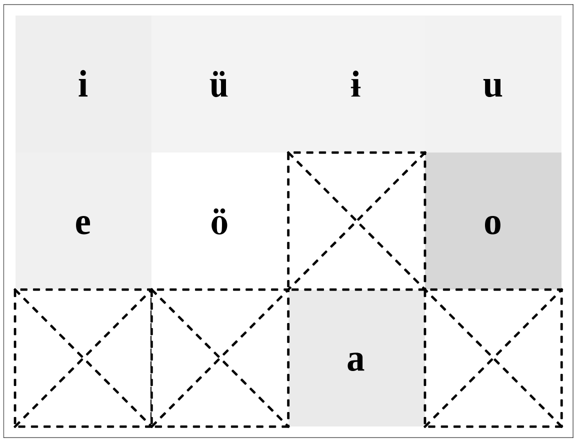


Figure 12: Tuvan Neutral Vowels

As a final note, it would be nice to know that the visualization tool can distinguish between languages with varying degrees of harmony. Turkish and Tuvan provide a good example of this, since they are easily comparable and Tuvan is known to be more strongly harmonic than Turkish.

A comparison of the visualizations for the two languages provides evidence that this tool can provide a visual interpretation of the strength of harmony, which is particularly clear when comparing the back vowel classes (Figure 13).

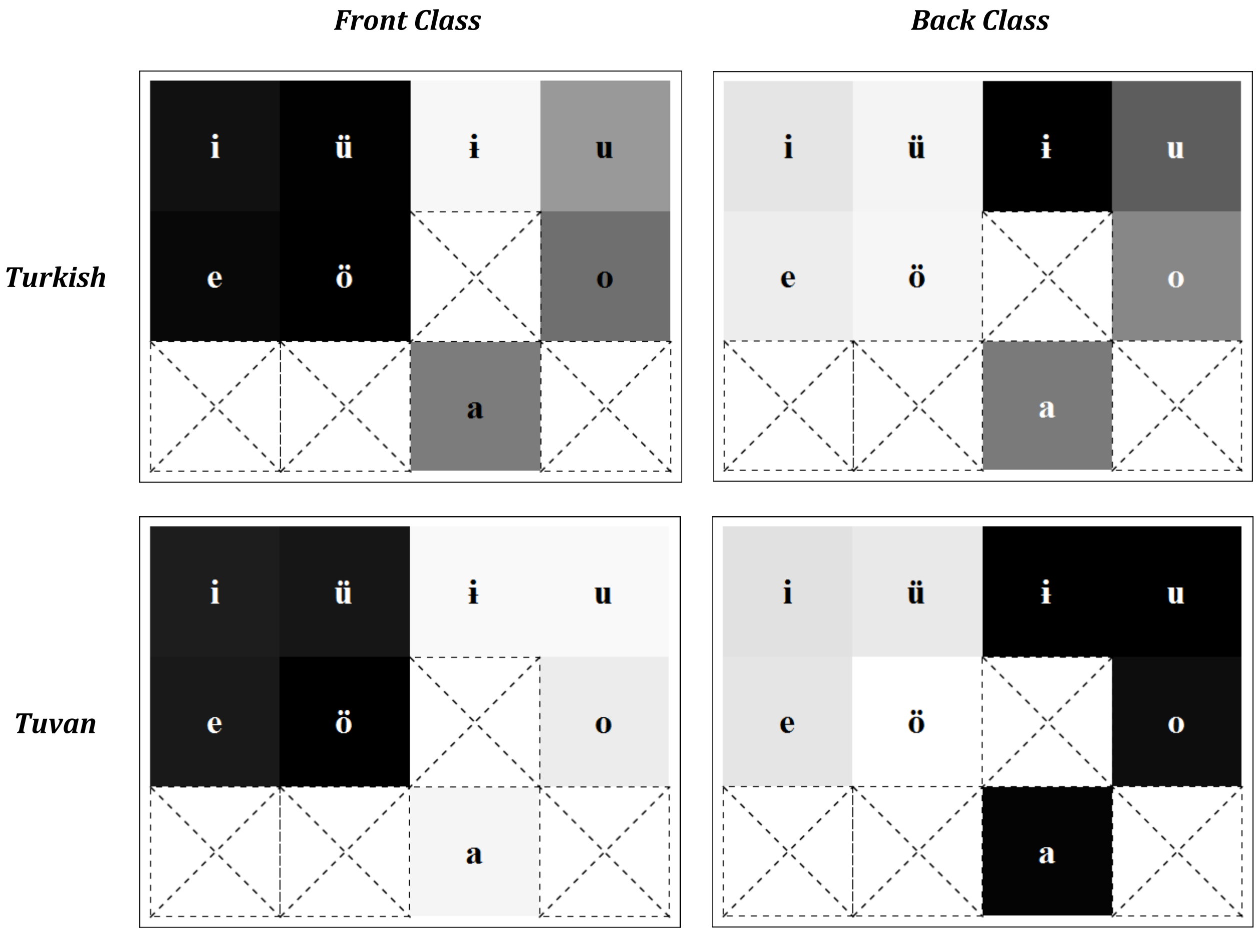


Figure 13: Visualization for Turkish and Tuvan Vowel Harmony

12 Conclusion and Future Work

In this thesis, I have shown that a simple Mixture of Unigrams model, which makes stricter whole-word harmony assumptions than a HMM, is capable of modeling vowel harmony systems in a way that coincides well with recent work on quantifying vowel harmony. While all of the results examined in Section 10 use Mixture-of-Two-Unigrams models, preliminary tests with Mixture-of-Four-Unigrams models suggest that it may also be possible to discover multi-dimensional harmony even with a model as simple as the Mixture of Unigrams. Through experiments on six languages (with varying levels of harmony), I have shown that what originally appear to be significant questions in data choice – types vs. tokens and monosyllables vs. no monosyllables – do not actually have as strong an impact on model output as one might expect. In addition to comparing across languages, these statistical tools for vowel harmony can be used, if given the appropriate data, to evaluate and quantify change in harmony over time. Seeing how and why languages become more or less harmonic could provide new and useful information about this phonological constraint.

In addition to exploring models of vowel harmony, I have presented a visualization tool that combines basic knowledge about a language's vowel inventory with the output of unsupervised models to produce graphics that can provide users with a fast preliminary way to make judgments on a language's vowel harmony system or lack thereof. This visualization tool can be used as an intermediate step to guide users to vowel harmony phenomena that may merit further investigation while also providing them with a convenient way to figure out which parameter settings may be appropriate if they wish to use another quantification tool like the VHC.

Given sufficient training data and appropriate machine learning techniques, Mixture of Unigrams model and HMMs could be used to directly choose vowel classes for the VHC parameters, thus cutting out the middleman and making the VHC more like an unsupervised model. The main benefit of this would be that, rather than having to test a language with an unknown harmony system by guessing what type of harmony it has or whether it has neutral vowels, the VHC (given only the list of vowels in the language) could determine the most appropriate harmony systems for which to test.

In the future, I'd like to see these kinds of tools for whole-word harmony combined with statistical methods for pairwise harmony, such as recent work by Sanders and Harrison (under review). A combination of these methods could lead to interesting research in the typology of vowel harmony with respect to what combinations of pairwise or whole word harmony and disharmony are attested in the world's languages.

APPENDIX I: AUTOMATICALLY SEPARATING VOWELS AND CONSONANTS

Depending on the dataset, its orthography or phonetic transcription style, and the user's knowledge thereof, it may be necessary to automatically separate vowels and consonants rather than using a list to extract the vowels in the corpus. In most cases, this will not be a concern, but showing that it is possible to do so automatically supports the unsupervised nature of the methods.

I.I SUKHOTIN'S ALGORITHM

Sukhotin's Algorithm (Sukhotin, 1962) was created by B. V. Sukhotin, a Soviet researcher, as an algorithm for identifying vowels in a simple substitution cipher. As the original article was produced in Russian then translated to French, the description here is based on Guy (1991). The algorithm makes the assumption that vowels occur next to consonants rather than other vowels. To determine what the vowels are in the word "llamado" using Sukhotin's algorithm, I first assume that I do not know where the word ends or begins (imagine that it is written in a circle), then I fill in a symmetric matrix with the number of times each letter in the word is adjacent to each other letter, then fill the diagonal with zeros:

Table 22: Table for Sukhotin's Algorithm

	L	A	M	D	O	*SUM*
L	0	1	0	0	1	2
A	1	0	2	1	0	4
M	0	2	0	0	0	2
D	0	1	0	0	1	2
O	1	0	0	1	0	2

Next, claim that the letter with the highest sum greater than zero is a vowel, and the rest are consonants. For each row of each consonant, subtract twice the number of times it occurs next to the new vowel from its sum:

Table 23: Updated Sukhotin's Algorithm Table

	L	A	M	D	O	*SUM*	
L	0	1	0	0	1	0	C
A	1	0	2	1	0	4	V
M	0	2	0	0	0	-2	C
D	0	1	0	0	1	0	C
O	1	0	0	1	0	2	C

Now repeat the previous portion, claiming that O is a vowel:

Table 24: Terminal Sukhotin's Algorithm Table

	L	A	M	D	O	*SUM*	
L	0	1	0	0	1	-2	C
A	1	0	2	1	0	4	V
M	0	2	0	0	0	-2	C
D	0	1	0	0	1	-2	C
O	1	0	0	1	0	2	V

Now that there are no letters left whose sum is greater than zero, the algorithm terminates (Guy, 1991: 259-260). If you experiment with short words, you will find that the algorithm does not always succeed, but its performance should improve given more data (assuming that the data does in fact have low probabilities of vowel-to-vowel transitions).

I.II Hidden Markov Models

Hidden Markov Models (described earlier) are "probabilistic finite state machines" (Baker, 2009: 10). A two state Hidden Markov Model that follows an alternating pattern can also be used to separate vowels and consonants into separate classes.

Appendix II: Other Models

This thesis has focused on three models for statistically modeling and quantifying whole-word harmony – the Mixture of Unigrams model introduced here, the HMM for its close relation to Mixture of Unigrams, and the VHC as a baseline against which to measure results. I would be remiss, however, were I to fail to provide at least a brief mention of some other related work on statistically modeling vowel harmony.

Goldsmith and Riggle (to appear), have explored information theoretic approaches including unigram, bigram, and Boltzmann models for vowel harmony in Finnish. Work from Baker (2009) builds on this and tests the models on a larger set of languages. Mailhot (2010) presents a regression-based model for harmony acquisition by learners. This type of model is based more heavily in psychology and perception, differentiating it from the models discussed in this thesis. Work by Mayer et. al. (2010) uses statistics on vowel successors to produce matrices visual displaying information about pairwise harmony and other harmony-like phenomena including reduplication and German umlaut. This thesis focuses on whole-word harmony rather than pairwise harmony, but the availability of visualization and quantification tools for both whole-word harmony and pairwise harmony sets the stage for interesting work on the typology of vowel harmony. Sanders and Harrison (under review) provide a statistical method for the quantification of pairwise harmony.

APPENDIX III: MORE ON MIXTURE OF UNIGRAMS

This section presents a more technical description of the Mixture of Unigrams model as applied to vowel harmony. First I introduce the pieces of the model:

Table 25: Notation for Mixture of Unigrams

Notation	**Description**
$V = \{0, \dots, m\}$	The set of vowels in the language, represented with integers for simplicity.
$Corpus = \{w_0, \dots, w_n\}$	The set of n words in the corpus.
$w_i = (v_0, \dots, v_{l-1})$	A word $w_i$in the corpus is made up of a set of l vowels, with $v_i \in V$ for $i \in \{0, \dots, l-1\}$. Word length l can be drawn from a multinomial with a Dirichlet prior, or another distribution, but this is not important when it comes to fitting the model.
$Classes = \{0,1\}$	The set of classes. For my purposes, I use two, but the model could accommodate more.
$Class[w_i]$	The class assigned to word w_i
$\theta \sim Dir(\sigma u)$	θ is a corpus-wide multinomial distribution over classes, drawn from a Dirichlet distribution with parameter vector σu where σ is the concentration parameter and u is the base measure.
$\psi_k \sim Dir(\delta m)$	For each class $k \in Classes$, ψ_k is a multinomial distribution over the set V drawn from a Dirichlet distribution with parameter vector δm where δ is the concentration parameter and m is the base measure.

Then the generative story is as follows:

- Draw $\theta \sim Dir(\sigma)$
- Draw $\psi_k \sim Dir(\delta)$ for each $k \in Classes$
- For each word $w_j \in Corpus$:
    - Draw $k = Class[w_j] \sim \theta$
    - Choose a length l for the word
    - For $i = 0, \dots, l-1$:
        - Draw $v_i \sim \psi_k$

It can also be represented using this plate notation. Each rectangular plate represents repetitions (the FOR loops in the generative story) and each disk represents a variable, with the colored disk representing the only observed variable. Here C stands for *Class*.

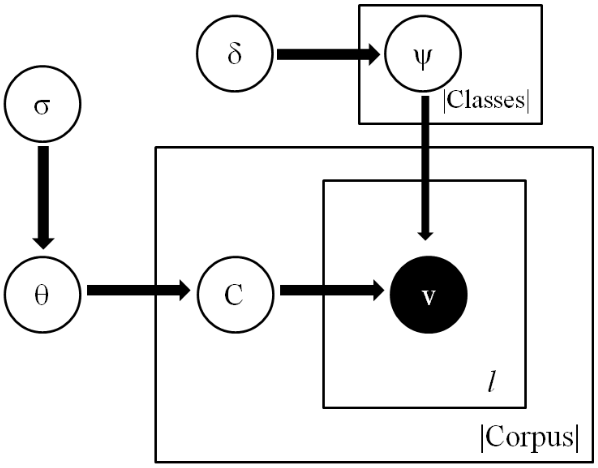


Figure 14: Plate Diagram for Mixture of Unigrams

III.I PSEUDOCODE FOR GIBBS SAMPLING

I begin the description of Gibbs sampling with a few notes on notation:

Table 26: Notation for Gibbs Sampling

Notation	**Explanation**
N_k	The count of the number of words in the corpus that are assigned to class k.
$N_k^{\setminus i} = \begin{cases} N_k, & i \neq k \\ N_k - 1, & i = k \end{cases}$	Where $i = Class[w_i]$. These are the corpus counts as though w_i were removed from the corpus.
$N_{j\|k}$	The count of the number of times the vowel indexed by j appears in words in the corpus that are assigned to class k.
$N_{j\|k}^{<h}, N_k^{<h}$	Let h index a vowel in w_i. Then $N_{j\|k}^{<h}$ and $N_k^{<h}$are the counts described above as though the vowels indexed h or greater in word $w_i$were removed from the corpus.
$Class[w_i]^{(t)}$	The class assigned to word $w_i$at iteration t.

Pseudocode:
(Based on work by Wallach, Knowles, and Dredze, presented in Knowles, 2011.)

```
*Initialization:*
Set $N_k = 0 \ \forall k \in Classes$
Set $N_{j|k} = 0 \ \forall k \in Classes, j \in V$
For $i = 0, \dots, n$:
    Initialize $Class[w_i] = k$ from the set of classes {0,1} randomly from a multinomial dist.
    Increment: $N_k += 1$
    For each vowel in $w_i$:
        Increment: $N_{vowel|k} += 1$
*Sampling:*
For $t = 0, \dots, I$ iterations:
    For $i = 0, \dots, n$:
        For each class $k \in Classes$:
            $\mathrm{Prod} = \frac{\sigma}{2} + \mathrm{N}_{\mathrm{k}}^{\setminus \mathrm{i}}$
            For each $j = 0, \dots, l - 1$:
                $\mathrm{Prod} = \mathrm{Prod} * \frac{\frac{\delta}{|V|} + \mathrm{N}_{\mathrm{v_j|k}}^{<j}}{\mathrm{N}_{\mathrm{k}}^{<j} + \delta}$
            $P(Class[w_i] = k) \propto \mathrm{Prod}$
        Draw $Class[w_i]^{(t+1)}$ based on the probabilities just defined
        Decrement counts: $N_k = \begin{cases} N_k, \ k \neq Class[w_i]^{(t)} \\ N_k - 1, \ k = Class[w_i]^{(t)} \end{cases}$
        For vowel in $w_i$:
            Decrement counts: $N_{j|k} = \begin{cases} N_{j|k} + 1, \ j = vowel \ and \ k = Class[w_i]^{(t)} \\ N_{j|k}, \ otherwise \end{cases}$
        Increment counts: $N_k = \begin{cases} N_k, \ k \neq Class[w_i]^{(t+1)} \\ N_k + 1, \ k = Class[w_i]^{(t+1)} \end{cases}$
        For vowel in $w_i$:
            Increment counts: $N_{j|k} = \begin{cases} N_{j|k} + 1, \ j = vowel \ and \ k = Class[w_i]^{(t+1)} \\ N_{j|k}, \ otherwise \end{cases}$
```

Appendix IV: Mixture of Unigrams Output

Table 27 and Table 28 contain visual output of vowel harmony systems and transparent vowels, respectively. Table 29 contains the raw output from mixture of unigrams run on each dataset with all combinations of tokens, types, monosyllables, and no monosyllables.

Table 27: Mixture of Unigrams Visualizations

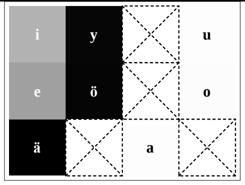
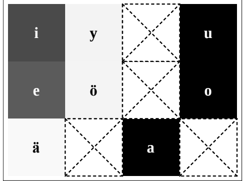
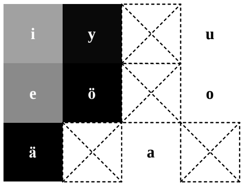
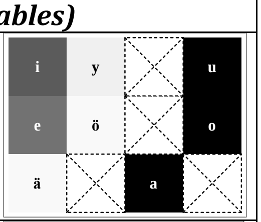
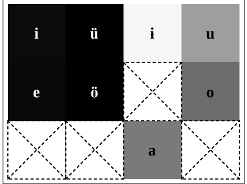
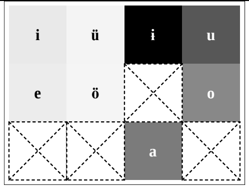
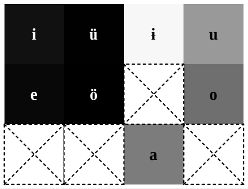
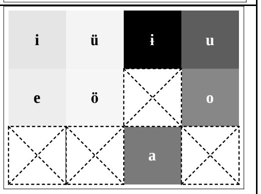
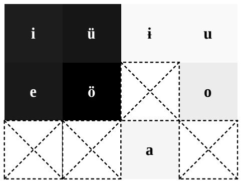
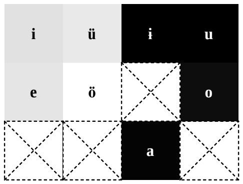
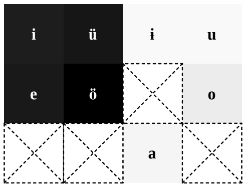
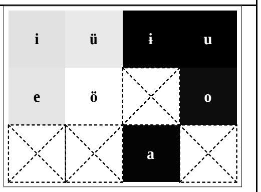
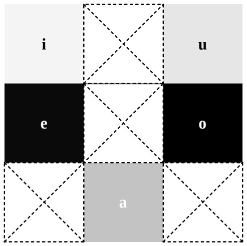
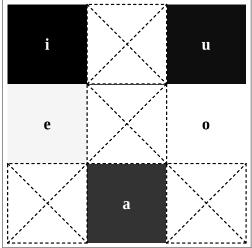
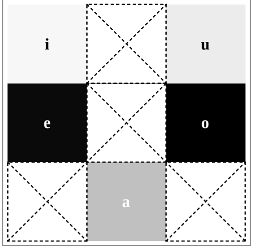
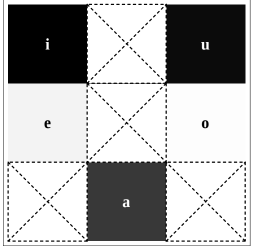
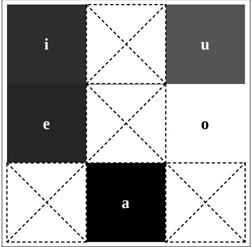
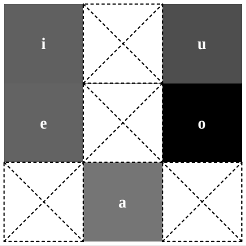
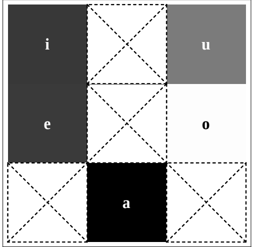
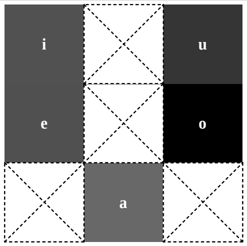
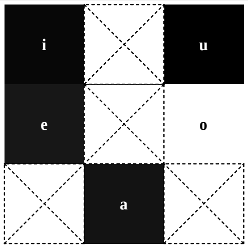
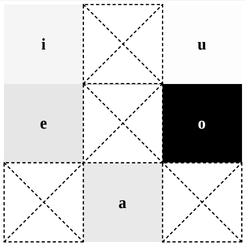
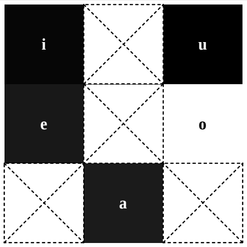
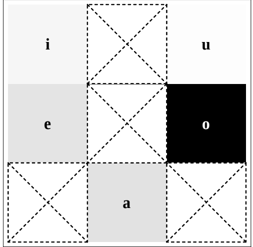
	Types (No monosyllables)		***Tokens (No monosyllables)***	
Finnish				
Turkish				
Tuvan				
Swahili				
Japanese				
Indonesian				

Table 28: Mixture of Unigrams Neutral Vowel Visualization

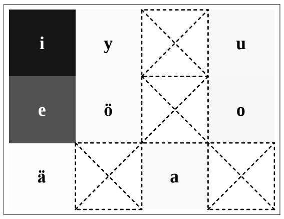
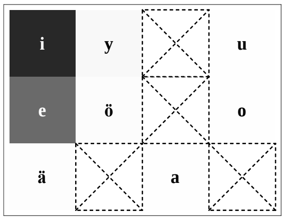
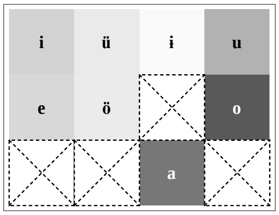
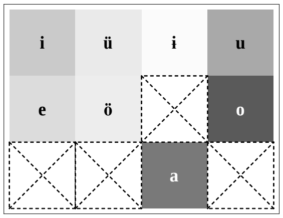
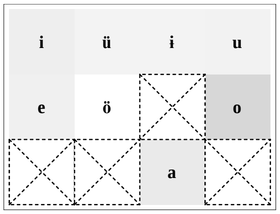
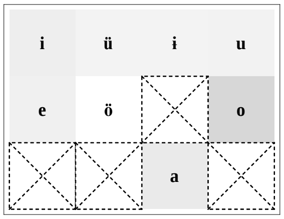
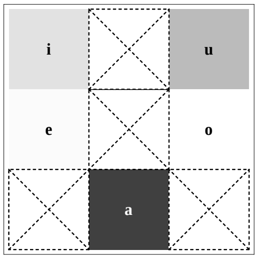
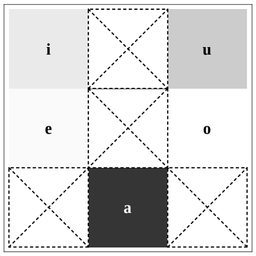
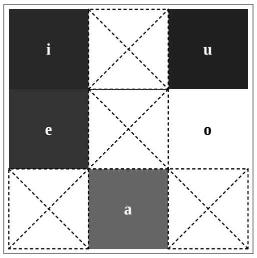
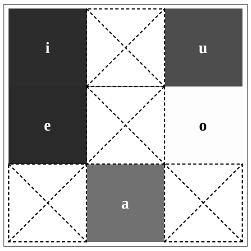
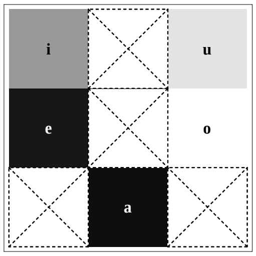
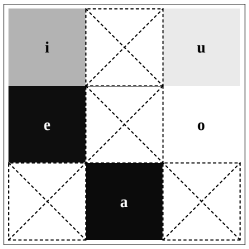
	Types (No monosyllables)	*Tokens (No monosyllables)*
Finnish		
Turkish		
Tuvan		
Swahili		
Japanese		
Indonesian		

Table 29: Vowel Probabilities from Mixture of Unigrams

Language	*Vowel*	*Tokens (No mono.) 0*	*Tokens (No mono.) 1*	*Types (No mono.) 0*	*Types (No mono.) 1*	*All Tokens 0*	*All Tokens 1*	*All Types 0*	*All Types 1*
Finnish	ä	0.004	0.359	0.355	0.003	0.004	0.359	0.355	0.003
	e	0.162	0.278	0.246	0.167	0.162	0.278	0.246	0.167
	i	0.217	0.257	0.227	0.208	0.217	0.257	0.227	0.208
	y	0.002	0.077	0.122	0.002	0.002	0.077	0.122	0.002
	ö	0.000	0.025	0.033	0.001	0.000	0.025	0.033	0.001
	a	0.322	0.002	0.009	0.352	0.322	0.002	0.009	0.352
	u	0.147	0.001	0.004	0.143	0.147	0.001	0.004	0.143
	o	0.146	0.001	0.004	0.123	0.146	0.001	0.004	0.123
Indonesian	a	0.424	0.406	0.416	0.441	0.424	0.406	0.416	0.441
	e	0.219	0.232	0.246	0.224	0.219	0.232	0.246	0.224
	i	0.057	0.192	0.071	0.178	0.057	0.192	0.071	0.178
	u	0.014	0.171	0.017	0.158	0.014	0.171	0.017	0.158
	o	0.286	0.000	0.250	0.000	0.286	0.000	0.250	0.000
Japanese	a	0.231	0.415	0.393	0.236	0.231	0.415	0.393	0.236
	i	0.219	0.266	0.240	0.202	0.219	0.266	0.240	0.202
	e	0.153	0.183	0.198	0.158	0.153	0.183	0.198	0.158
	u	0.192	0.134	0.168	0.193	0.192	0.134	0.168	0.193
	o	0.205	0.002	0.000	0.210	0.205	0.002	0.000	0.210
Swahili	a	0.457	0.362	0.462	0.345	0.457	0.362	0.462	0.345
	e	0.007	0.332	0.006	0.326	0.007	0.332	0.006	0.326
	o	0.001	0.238	0.000	0.241	0.001	0.238	0.000	0.241
	u	0.197	0.040	0.180	0.048	0.197	0.040	0.180	0.048
	i	0.338	0.028	0.352	0.040	0.338	0.028	0.352	0.040
Turkish	a	0.249	0.474	0.472	0.251	0.249	0.474	0.472	0.251
	ɨ	0.004	0.228	0.229	0.004	0.004	0.228	0.229	0.004
	u	0.040	0.118	0.124	0.038	0.040	0.118	0.124	0.038
	o	0.050	0.077	0.077	0.050	0.050	0.077	0.077	0.050
	i	0.250	0.053	0.044	0.253	0.250	0.053	0.044	0.253
	e	0.306	0.042	0.046	0.302	0.306	0.042	0.046	0.302
	ü	0.081	0.007	0.007	0.081	0.081	0.007	0.007	0.081
	ö	0.020	0.001	0.002	0.020	0.020	0.001	0.002	0.020
Tuvan	a	0.034	0.407	0.036	0.439	0.034	0.407	0.036	0.439
	ɨ	0.012	0.245	0.014	0.264	0.012	0.245	0.014	0.264
	u	0.009	0.172	0.013	0.121	0.009	0.172	0.013	0.121
	o	0.020	0.125	0.018	0.088	0.020	0.125	0.018	0.088
	e	0.419	0.025	0.420	0.046	0.419	0.025	0.420	0.046
	i	0.301	0.020	0.292	0.029	0.301	0.020	0.292	0.029
	ü	0.129	0.006	0.130	0.013	0.129	0.006	0.130	0.013
	ö	0.076	0.000	0.077	0.000	0.076	0.000	0.077	0.000

Sources Consulted:

Adams, D. 1980. *The Hitchhiker's Guide to the Galaxy*. New York: Harmony Books.

Baker, A. 2009. Two Statistical Approaches to Finding Vowel Harmony. Technical Report. University of Chicago. http://www.cs.uchicago.edu/research/publications/techreports/TR-2009-03

Baković, E. 2000. *Harmony, Dominance and Control*. Ph. D. Diss. New Brunswick, NJ: Department of Linguistics, Rutgers University.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss. 1970. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov chains. *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171.

Beckman, J. N. 1997. Positional Faithfulness, Positional Neutralisation and Shona Vowel Harmony. *Phonology* 14. Cambridge University Press, pp. 1-46.

Blei, D., A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, pp. 993–1022, January.

Carnegie Mellon University. CMU Pronouncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Childs, G. T. 2003. *An Introduction to African Languages*. John Benjamins Pub. Co. http://books.google.com/books?id=2wMX6tJQFBMC&lpg=PP1&dq=inauthor%3A%22George%20Tucker%20Childs%22&pg=PP1#v=onepage&q&f=false

Clements, G.N. 1976. The Autosegmental Treatment of Vowel Harmony. In W.U. Dressler and O. Pfeiffer (eds.), *Phonologica*, Innsbrucker Beiträge zur Sprachwissenschaft vol. 19), pp. 111-119.

Clements, G.N. and E. Sezer. 1982. Vowel and Consonant Disharmony in Turkish. In Hulst, H. van der and N. Smith (eds.). *The Structure of Phonological Representations: Part II*, Dordrecht-Holland: Foris, pp.213-356.

Ellison, T. M. 1991 The Iterative Learning of Phonological Constraints. Computational Linguistics, Vol. 20. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.8768&rep=rep1&type=pdf

Foster, M. 2011. La.'s Tunica Tribe Revives its Lost Language. ABC News. Aug. 6. Accessed Oct. 19, 2011. http://abcnews.go.com/US/wireStory?id=14246211

Goldsmith, J. 1976. *Autosegmental phonology.* Ph. D. Diss. Massachusetts Institute of Technology.

Goldsmith, J. and A. Xanthos. 2009. Learning Phonological Categories. Language, Vol. 85, Num. 1. March, pp. 4-38. *Project MUSE*. Web. 21 Jan. 2011. http://muse.jhu.edu/journals/lan/summary/v085/85.1.goldsmith.html

Goldsmith, J. and J. Riggle. (To appear). Information Theoretic Approaches to Phonology: The Case of Vowel Harmony. To appear in *Natural Language and Linguistic Inquiry*.

Guy, J. B. M. 1991. Vowel Identification: An Old (but Good) Algorithm. Cryptologia, Vol. XV, Num. 3. July. http://languagelog.ldc.upenn.edu/myl/guySukhotin.pdf

Hall, T A. 1999. The Phonological Word: A Review. *Studies on the Phonological Word*. Ed. Hall, T A, and Ursula Kleinhenz. Amsterdam: J. Benjamins.

Harrison, K. D., E. Thomforde and M, O'Keefe. 2004. The Vowel Harmony Calculator. http://www.swarthmore.edu/SocSci/harmony/public_html/index.html

Heeringa, W. and A. Braun. 2003. The Use of the Almeida-Braun System in the Measurement of Dutch Dialect Distances. Computers and the Humanities Vol. 37, No. 3, Computational Methods in Dialectometry, August, pp. 257-271. http://www.jstor.org/stable/30204901

Hulst, H.G. van der and J. van der Weijer. 1995. Vowel harmony. In: J.A. Goldsmith (ed.). *The Handbook of phonological theory*. Basil Blackwell, Oxford, pp. 495-534. Accessed online from: http://linguistics.uconn.edu/People/vanderhulst.html

Jurafsky, D. and J. H. Martin. 2009. *Speech and Language Processing, 2nd Ed*. Upper Saddle River, NJ: Pearson Prentice Hall.

Katamba, F. 1989. *An Introduction to Phonology*. New York: Pearson Education.

Knowles, R. L. 2011. Register Topic Models. (Work with H. Wallach and M. Dredze) Mid-Atlantic Student Colloquium on Speech, Language and Learning. Johns Hopkins University. September 23.

Krämer, M. 2003. *Vowel Harmony and Correspondence Theory*. Berlin: Mouton de Gruyter. Internet resource.

Ladefoged, P. 2005. *Vowels And Consonants, An Introduction To The Sounds Of Languages*. Wiley-Blackwell.

Lewis, M. P. (ed.). 2009. *Ethnologue: Languages of the World, Sixteenth edition*. Dallas, Tex.: SIL International. Online version: http://www.ethnologue.com/

Mailhot, F. 2010. Instance-based Acquisition of Vowel Harmony. In J. Heinz, L. Cahill and R. Wicentowski (eds.) *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, Uppsala, Sweden, July, pp.1-8. http://www.aclweb.org/anthology/W10-2201

Mayer, T., C. Rohrdantz, M. Butt, F. Plank, D. A. Keim. 2010. Visualizing Vowel Harmony. *Linguistic Issues in Language Technology*. CLSI Publications, December.

Prince, A. and P. Smolensky. 1993, 2002. Optimality Theory: Constraint Interaction in Generative Grammar. *ROA*.

Rabiner, L. R. and B. H. Juang. 1986. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine,* Vol. 3, No. 1, January, pp. 4-16.

Rabiner, L. R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE,* Vol. 77, No. 2, February, pp. 257-286.

Resnik P. and E. Hardisty. 2010. Gibbs Sampling for the Uninitiated. Technical Report. University of Maryland, April.

Ringen, C. O. and O. Heinämäki. 1999. Variation in Finnish Vowel Harmony: An OT Account. *Natural Language & Linguistic Theory,* Volume 17, Number 2. May.

Sanders, N. and K.D. Harrison. Under review. Discovering New Vowel Harmony Patterns Using a Pairwise Statistical Model.

Smolek, A. 2011. Vowel Harmony in Tuvan and Igbo: Statistical and Optimality Theoretic Analyses. Undergraduate Thesis, Swarthmore College. http://www.swarthmore.edu/SocSci/Linguistics/2011thesis/PDFs/Smolek.pdf

Sukhotin, B. V. 1962. Exsperimental'noe Vydelenie Klassov Bukv s Pomoščju EVM. *Problemy strukturoj lingvistiki,* 234, pp. 189-206.

Wallach, H., R. L. Knowles, and M. Dredze. 2011. Unpublished code and notes for work on Register Topic Models.

Wolfram Research, Inc. 2010. Mathematica. Version 8.0. Champaign, IL.