Vowel Harmony Viewed as Error-Correcting Code

Yvo Meeres
University of Hamburg
Hamburg, Germany
yvo.meeres@mailbox.org

Tommi A Pirinen
UiT Norgga árktalaš universitehta
Hansine Hansens Veg 16
Tromsø, Noreg
tommi.pirinen@uit.no

Abstract

Robustness reduces the risk of information loss. At present the notion of error-correcting codes (ECCs) is used to achieve robustness in technical fields only. Viewing fault-tolerant natural systems as systems equipped with error-correcting codes permits a formal comparison of natural and technical robustness. Instancing natural language (NL), we show differences in technical and natural error-correcting approaches. By picking a specific grammar phenomenon which some NLs exhibit – vowel harmony (VH) – we show that (1) VH can be formalized as an ECC as well as (2) VH adds to the robustness of its NL. We provide empirical as well as formal evidence on this fact. (3) Consequently, the example of VH shows that the notion of an ECC serves as a suitable formal model not only for technical but also for natural robustness.

1 Introduction

The phonological grammar rules of vowel harmony can be formalized in a way reflecting their ability to contribute to the robustness of a natural language (NL). This paper combines the field phonology as a subfield from linguistics with the field coding theory which is part of the engineering disciplines and mathematics. This new combination permits a concise mathematical comparison of technical and natural robustness. Why? Since NL is one example of a fault-tolerant system in nature, picking one phonological grammar phenomenon suffices to show that a formal, mathematical comparison is possible. Robust systems trigger a wide range of research and may profit from a united formal model.

First the dataset, methods and notations are provided in Section 2. After formalizing vowel harmony (VH) in Section 3 and error-correcting-codes (ECCs) in Section 4, the combination of the two in Section 5 allows to provide both mathematical as well as empirical evidence that VH acts as an ECC in Section 6. The NLs Finnish and Turkish serve as running examples, abbreviated as `fi` and `tr`, respectively.

Vowel harmony (Krämer, 2003; Archangeli and Pulleyblank, 2007) is a linguistic grammar phenomenon situated in the phonological layer. It restricts the possibilities of vowels that can appear within a scope. Since VH does not convey semantic information, the purpose of this grammar phenomenon is not obvious. We refer the reader to (Lloret, 2008) which provides an overview of literature on the purpose of VH. Henriksen (2017) provides empirical evidence for more reliable discrimination with VH than without. Kimper (2017) provides an elaborate literature review as well as new empirical evidence on the benefit of VH during perception. Artificial grammar learning experiments show that speakers of languages not exhibiting VH can acquire a presented harmony easily (Pycha et al., 2003; Moreton and Pater, 2012; Finley and Badecker, 2008; Baer-Henney et al., 2015). Complexity considerations link learnability to a fine-grained subregular categorization (Heinz and Lai, 2013; Hwangbo, 2015). All in all, due to this vivid and thorough research, VH offers itself as an example for comparing robustness in natural and technical systems. On top of that, VH is a semantic-free grammar phenomenon meaning that it is not designed to transmit semantic information by itself – perfect for our aim of analyzing the error-correcting capabilities of VH, since no semantic purpose of the grammar phenomenon causes interferences. Choosing instead plural formation or inflection as subject of study would exactly present us with this problem.

Error-correcting codes (ECCs) aim at transmitting information in a robust but at the same time efficient way. The raw information is not transmitted. Instead, the information is first content enriched with redundancy to enable the detection and correction of transmission errors. Although the basics of coding theory have been solved in an efficient way, still new research questions pop up when adapting ECC to new contexts. For example, ECC for Mud Pulse Telemetry (MPT) (Mwachaka et al., 2019) and for the 5G mobile network (Mansoor and Ismaeel, 2019; Wonterghem et al., 2018; Dubrova et al., 2016) received interest in current literature. ECCs have been picked up as a method itself for hardness amplification in complexity theory (Arora and Barak, 2009; Dwork et al., 2009; Sudan et al., 1999). The field dealing with ECCs is called *coding theory*. The algebraic (Berlekamp, 1968) perspective prevails here and was the starting point (Hamming, 1950). Viewing channel codes from an automata and formal language theory's perspective is rare but exists (Marchenko et al., 2018; Zavadskyi, 2015; Anisimov and Zavadskyi, 2014). Formal languages have proven useful for processing of NL and therefore impose themselves for ECC modeling in our scenario. The technical applications for ECCs are manifold. But ECCs do not necessarily be constructed. They already occur in nature. DNA is a robust and fault-tolerant storage medium (Brady et al., 2009) to mention only one example. Hence, we provide a definition of an ECC which is neither limited to technical applications nor to the classical methods for ECC-construction.

2 Preliminaries

2.1 Data

For the homonym experiment in Subsection 6.1 seldom lemmata are of significant importance since we would lose potential homonyms otherwise. For this reason we chose the huge corpus of `omorfi` (Pirinen, 2015)[1], since it is incorporating as many lemmata as possible to circumvent parsing errors. However, again for circumventing parsing errors, this corpus includes a high percentage of proper nouns and non-Finnish terms. In the first step, these are filtered out. From the freely available[2] word embeddings (TurkuNLP, 2019) the remaining list of lemmata receives its `word2vec` distances (Luotolahti et al., 2015; Zeman et al., 2018). Our scripts and data from the conducted experiment are available[3] including among others the filter script, the `word2vec` processing and our manual classification.

2.2 Methods

To answer the research question of whether natural robust systems can be viewed as ECCs, word embeddings as well as notions from coding and automata theory provide the methods. The mathematical fundament from coding theory will be ported or adapted to natural systems. Since we choose vowel harmony as an example for a natural system, the methods used in automata and formal language theory suit the purpose. Since coding theory deals with word distances, the method `word2vec` (Mikolov et al., 2015) is useful.

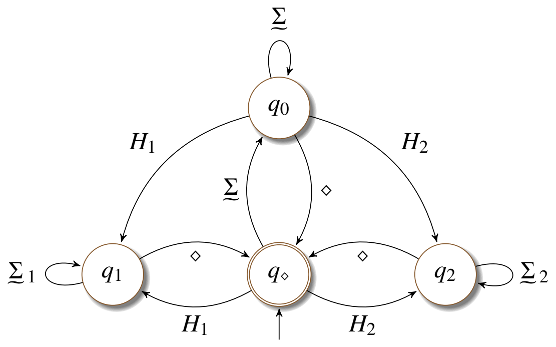
2.3 Notation

An *alphabet* Λ is a finite set of *symbols*. Symbols added to an alphabet are given in the subscript: $\Lambda_\diamond = \Lambda \cup \{\diamond\}$. Λ^* denotes the set of all strings over Λ. Such a *string* $s \in \Lambda^*$ will be called a *word* or an *utterance* depending on the context. The length of a string s is written as $|s|$. Σ will denote an alphabet with terminal symbols only. On the contrary, Γ denotes an alphabet which comprises both terminals and nonterminals. We call a symbol from an alphabet a nonterminal, if it does not occur when pronouncing a NL, i.e. if it is not phoneme and thus not *pronounceable*. For a NL the phonemes Σ are divided into the disjunct alphabets of vowels V and consonants C. H is an alphabet of nonterminals called *abstract vowels* which are mapped during language production by the vowel harmony into *pronounceable vowels* V. B is the *alphabet of delimiters*. None of the delimiters are phonemes: $B \cap \Sigma \neq \emptyset$. All kind of delimiters for NL are elements of B: pause in spoken language, space and interpunctuation in written language. In the course of the paper, strings do not comprise those delimiters; they are eliminated. The only delimiters occurring are *delimiters for VH* $\diamond, \natural \in B$. The power set is denoted by $P(\cdot)$. Finite state automata (FSA) and transducers are defined in the usual way. An FSA is a quintuple $(Q, \Lambda, q_0, \delta, F)$ with states Q, a start state q_0, a partial transition function δ and final states $F \subseteq Q$. A metric $\Delta : \Lambda^* \times \Lambda^* \to \mathbb{R}$ is a measure for a distance between two strings with

[1] `omorfi` in version 20191111 https://github.com/flammie/omorfi/releases/tag/20191111

[2] The `word2vec` embeddings are available at https://github.com/jmyrberg/finnish-word-embeddings.

[3] The git-repository for our conducted experiment is https://codeberg.org/vhecc/experiment-minpair.



$\underset{\sim}{\Sigma}$
q_0
H_1
H_2
$\underset{\sim}{\Sigma}$
$\diamond$
$\underset{\sim}{\Sigma}_1$
q_1
$\diamond$
$q_\diamond$
$\diamond$
q_2
$\underset{\sim}{\Sigma}_2$
H_1
H_2

Figure 1: The Finnish Vowel Harmony Automaton fiVHA

H_0	H_1	H_2	H
e	a	ä	$\textcircled{\dot{a}}$
i	o	ö	$\textcircled{\dot{o}}$
	u	y	$\textcircled{\dot{u}}$

(a) Finnish

H_{1a}	H_{1b}	H_{2a}	H_{2b}	H
e	a	o	ö	$\textcircled{\substack{ea\\ o\ddot{o}}}$
i	ı	u	ü	$\textcircled{\substack{i\,\imath\\ u\ddot{u}}}$

H'_1	H'_2	H'
e	a	$\textcircled{\substack{e\\ a}}$
i	ı	$\textcircled{\substack{i\\ \imath}}$
o	ö	$\textcircled{\substack{o\\ \ddot{o}}}$
u	ü	$\textcircled{\substack{u\\ \ddot{u}}}$

(b) Turkish

Table 1: The vowel harmonies with their corresponding abstract vowels

(1) $\Delta(x, y) = 0 \Leftrightarrow x = y$, (2) $\Delta(x, y) = \Delta(y, x)$ and (3) *triangle inequality*: $\Delta(x, y) \le \Delta(x, z) + \Delta(z, y)$.

3 Vowel harmony (VH)

Vowel harmony ensures the exclusion of certain vowels in a vowel harmony's scope. The allowed vowels within a scope are chiming together, to say it in a poetic way. After the definition of VH classes, we can define VH from a formal language theoretic perspective.

Definition 1 (Vowel Harmony Classes)**.** A vowel *harmony class* H_j with index $j \in h$ is a subset of the pronounceable vowels V. Let h be the set of indices of the harmony classes, where h is a sequence starting from 0 or 1 respectively depending on whether the vowel harmony at hand has a neutral class or not. Thus the *neutral vowel harmony class* is denoted by H_0. A tilde above a symbol indicates the exclusion of neutral vowels; a tilde below the exclusion of the nonneutral vowels: the set of indices for the nonneutral vowels is written as $\tilde{h} = h \setminus \{0\}$, the *nonneutral vowels* as $\tilde{V} = \cup_i H_i, \forall i \in \tilde{h}$. Additionally to $\underset{\sim}{\Sigma} = \Sigma \setminus \tilde{V}$, an index $i \in \tilde{h}$ indicates that a harmony class is added $\underset{\sim}{\Sigma}_i = \underset{\sim}{\Sigma} \cup H_i$. Observe that $\underset{\sim}{\Sigma} = C \cup H_0$.

At least the harmony sets H_1 and H_2 thus always exist. H_0 occurs often in a NLs harmony. For example, Finnish exhibits the harmony class H_0. Turkish, on the other hand, has no neutral vowels H_0.

Definition 2 (Vowel Harmony)**.** A *vowel harmony* is a tuple $(\mathscr{H}, V, \upsilon, \diamond)$, for short just $\mathscr{H}$. It arranges all its pronounceable vowels by $\upsilon : V \to \mathscr{H}$ into its harmony classes $H_i \in \mathscr{H}$ with $\forall v(v \in V \Rightarrow \exists i : v \in H_i)$ where $i \in h$ and imposes the requirements that all harmony classes are nonempty and $|\tilde{h}| > 1$. A symbol $\diamond \in B$ serves as delimiter for $\mathscr{H}$.

This means that a minimum of two nonneutral harmony classes must exist. We can construct the terminal set of vowels from the harmony classes: $V = \cup_j H_j,\ \forall j \in h$. A VH yields its *VH language*.

Definition 3 (Vowel Harmony Language (VHL))**.** The (proper) *vowel harmony language* VHL over the harmony $\mathscr{H}$, with $\mathsf{VHL} \subseteq \Lambda^*_\diamond$, is given by:

$$\mathsf{VHL}(\Lambda_\diamond) = \{\, (w\diamond)^* \mid w \in \underset{\sim}{\Lambda}^*_j\,,\ j \in h\,,\ \diamond \in B \,\}.$$

A $\mathsf{VHL}(\Lambda_\diamond)$ with $V \not\subseteq \Lambda$ is given by $\mathsf{VHL}(\Lambda_\diamond) = \{\, (w\diamond)^* \mid w \in \Lambda^* \,\}$. In a *VH utterance* $u \in \mathsf{VHL}$ a symbol $\diamond \in B$ denotes the end of every *VH word* $w \in \underset{\sim}{\Lambda}^*_j$ for all $j \in h$ to indicate the scope of the vowel harmony. For brevity a(n) VH utterance / word is just called a(n) utterance / word if the context is clear. An utterance *u complies* with a harmony iff $u \in \mathsf{VHL}$.

The above formula for VHL shows that a VH thus just draws its words from different alphabets, since the scope of one harmony class is exactly one VH word.

Example 1 (Finnish Vowel Harmony)**.** *Table 1a shows* fi$\mathscr{H}$*, the Finnish VH with its three harmony classes, the neutral one* H_0*, as well as* H_1 *and* H_2*. The Finnish VH language* $\mathsf{fiVHL} = \{\, (w\diamond)^* \mid w \in \underset{\sim}{\Sigma}^*_1 \cup \underset{\sim}{\Sigma}^*_2 \,\}$ *complies with the Finnish vowel harmony but to no other grammar phenomena. Thus, it comprises also utterances which convey no semantics. The transition function* δ *given by Figure 1 illustrates the DFA for recognizing* fiVHL*. The omitted dead state would indicate a harmony clash.*

Finnish is very strict in its vowel harmony. Compounds are written without a space, thus the $\diamond$ is compressed away here, but apart from that every lemma complies with vowel harmony. Even loan words are either changed for compliance or, if not, at least in spoken language often pronounced 'wrongly' – that is, not realized as written – but right in terms of the Finnish VH. In that last case only the spoken version complies with VH. In languages not that strict, like Turkish, it sometimes needs a delimiter within a Turkish semantic entity, c.f. Example 2.

Corollary 1 (VH Compliance)**.** Let $u, u' \in \Lambda_\diamond^*$ be utterances. Every utterance $u \notin \mathsf{VHL}$ can be adapted to an $u' \in \mathsf{VHL}$ in order to comply with a VH without changing the vowels occurring in u.

Proof. This follows trivially from Definition 3: Utterances contradicting VH, thus $u \notin \mathsf{VHL}$, are constructed by drawing vowels v_k, v_l from different harmony classes H_k, H_l, resp. and place them adjacent or with no delimiter in between. Separating those contradicting vowels at an arbitrary position by a $\diamond \in B$ makes u' an element of VHL. □

For certain constructions Corollary 1 is helpful. In the Turkish language, some lemmata comply with neither of Turkish's vowel harmonies. For example, we would need to write *el⋄ma⋄* (apple) and *an⋄ne⋄* (mother), thus using a VH delimiter inside a semantic entity.

Example 2 (Turkish Vowel Harmony)**.** *Turkish splits its vowels twice into a harmony, see them listed in Table 1b. The eight vowels are split into two classes by the harmony* $\mathsf{tr}\mathscr{H}'$*, those two sets are split again resulting in four harmony classes for harmony* $\mathsf{tr}\mathscr{H}$*. This yields a nested VH where inside a VH word ended with* $\diamond$ *for* $\mathsf{tr}\mathscr{H}'$ *a second delimiter* ♮ *may occur for* $\mathsf{tr}\mathscr{H}$*, e.g.* an⋄neciğim♮ *(mother,* diminutive*).*

4 Error-Correcting Code (ECC)

Error-correction seeks to improve robustness over a noisy channel at the expense of adding redundancy. A broader definition for ECCs than the one used in literature (Arora and Barak, 2009, 19.2.1 and 19.2.3) reflects also the character of non-technical codes. Imagine, you want to retrieve specific information from some data. The use-case of technical error-correction is the reconstruction of the whole data. Natural systems often have the use-case to retrieve specific information in contrary to reconstruct the entire data stream. List decoding (Zhang et al., 2020) goes slightly into this direction by delivering not one but several decoded candidates. We view a function as an ECC concerning a specific noise type with respect to the requested information, if and only if we can retrieve the information (partially) more often and better (see metric below) if preprocessed with ECC before exposure to the noise than the pure data exposed to the same type of noise. We formalize the preceding notion of error-correction as follows.

Definition 4 (Error-Correcting Code (ECC))**.** Let ECC be computable and Γ_{raw} and Λ_{ecc} alphabets for

$$\mathsf{ECC} : \Gamma_{\mathsf{raw}}^* \to P(\Lambda_{\mathsf{ecc}}^*).$$

In $c \in \mathsf{ECC}(w)$ we call $w \in \Gamma_{\mathsf{raw}}^*$ a *raw word* and $c \in \Lambda_{\mathsf{ecc}}^*$ its *code word*. Let Δ be a metric over $\Lambda = \Gamma_{\mathsf{raw}} \cup \Lambda_{\mathsf{ecc}}$. Let $\cdot' : \Lambda^* \to P(\Lambda^*)$ be a channel type (*noise*) with c' and w' called the corrupted versions of c and w, resp. We extend it to sets by applying it to every element of the set. Computable decoding functions $i_. : \Lambda \to I$ retreive the information $i \in I$ announcing in their indice for which input type they retreive the information. ECC is called an *error-correcting code* (ECC) with respect to information i and channel $'$ iff

$$\sum_{w \in \Gamma_{\mathsf{raw}}^*} \sum_{c' \in \mathsf{ECC}(w)'} \Delta\left(i_{c'}(c'), i_w(w)\right) - \Delta\left(i_{w'}(w'), i_w(w)\right) > 0.$$

To sum it up, the ECC should be advantageous to the raw version with respect to both the requested information and the type of noise. This is a very broad definition to capture all use-cases of error-correcting capabilities.

Technical error-correcting codes impose additional constraints:

§1 The raw word w is shorter than the code word c (called *block length*): $\Lambda^{|w|} \to \Lambda^{|c|}$ with $|w| < |c|$.

§2 The input and the output alphabets are identical: $\Lambda = \Gamma_{\mathsf{raw}} = \Lambda_{\mathsf{ecc}}$.

§3 The function ECC delivers only one code word: $\Lambda^* \to \Lambda^*$ and not possibly a set of code words.

§4 Arbitrary input w is encoded – in contrary above definition returns the empty set for unvalid w.

§5 The metric operates directly on the symbols, like edit distances (Winter et al., 2020) or the Hamming distance, and not on their semantics.

5 Vowel Harmony ECC (VHECC)

Viewing vowel harmony as an error-correcting code results in a definition for a vowel harmony error-correcting code (VHECC). The combination of VH, as defined in Section 3, with the generalized definition of an ECC, as given in Section 4, will allow us in Section 6 to formally and concisely analyze the capabilities of vowel harmony for detecting errors and partly correcting them. This section provides three definitions of ECCs for VH: (1) a general one, (2) one which historically formed the lexicon and (3) a real-time ECC where the VH operates on the morphology.

Definition 5 (Vowel Harmony Error-Correcting Code (VHECC))**.** Let $\Gamma = \Sigma \cup H$. The vowel harmony error-correcting code VHECC over a harmony $\mathscr{H} = \left(\mathscr{H}, H_0 \cup (H \times \tilde{h}), \upsilon, \diamond\right)$ is an ECC (see Def. 4) with

$$\mathsf{VHECC} : \Gamma^*_\diamond \to P(\Lambda^*_\diamond)$$

which maps an utterance u to all combinations of code words where every code word $c \in \mathsf{VHL}$ and where every abstract vowel $a \in H$ is either mapped to itself or to one of its vowels $v = (a, i)$ for $i \in \tilde{h}$. A VHECC is called *complete* if the input alphabet does not contain pronounceable nonneutral vowels and the output alphabet contains only terminals: $\mathsf{cVHECC} : \utilde{\Gamma}^*_\diamond \to P(\Sigma^*_\diamond)$.

Note that here for VHECC we need the broad notion of an ECC, c.f. Def. 4. In several aspects the technical ECCs differ. First of all, the length of a VH word as well as of a whole utterance remains the same when VHECC has been applied, in contrast to technical ECCs see §2 on page 4. Secondly, this is only possible by enlarging the alphabet, see §2. Astonishingly, VHECC uses the power set of utterances as codomain, since different code words for the same raw utterance are allowed in contrary to one codeword per raw word in §3.

Example 3 (Finnish VHECC)**.** *The partitioning of V for* fiVHL *into harmony classes (recall Example 1) is extended to a $H \times \tilde{h}$-matrix shown in Table 1a. This requires the nonneutral VH classes H_i to be of same size to permit a bijection. The abstract vowels H are ⓐ, ⓞ and ⓤ. Every vowel $v \in V$ is then a tuple of its abstract vowel and its harmony class (a, i) with $a \in H, i \in \tilde{h}$. The complete VHECC for Finnish can be expressed nicely by a nondeterministic transducer, c.f. Figure 2a. All abstract vowels $a \in H$ are nondeterministically replaced by their pronounceable counterparts.*

Example 4 (Turkish VHECC)**.** *Since Turkish exhibts no neutral class, the vowels are mapped into* 2×4 *and* 4×2 *matrices, see Table 1b. This is astonishing since the number of vowels, a quite central quantitiy for a language, has to be a product of natural numbers in the nonneutral VHECC case, as here for Turkish.*

These complete VHECCs turn out to have two components: the lexicon of a NL was constructed with a VHECC and language production, thus speaking or writing, performs a VHECC on the fly.

5.1 Lexical VHECC

VH has a historical dimension. NLs exhibiting VH constructed their lemmata complying (partly) to VH. The span of a VH word often coincides with a lemma. The power set in the following definition is needed only in the rare cases of preVH homonyms which we will treat in Subsection 6.1. Historically the language 'decided' which harmony a lemma should have. The function xVHECC can thus only be specified by listing all nondeterminstic historical choices the language has made. The x stands for the x in lexicon:

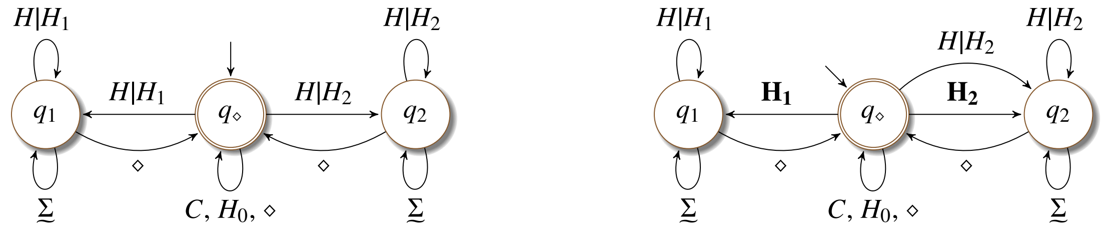
Definition 6 (Lexical VHECC; Lemma; Lexicon)**.** The *lexical* vowel harmony error-correcting code is a VHECC

$$\mathsf{xVHECC} : \utilde{\Gamma}^*_\diamond \to P(\Gamma^*_\diamond).$$

The union over all code words results in a set of *lemmata* with those lemmata drawn from Γ^* or $\Gamma^*_\diamond$. The set of lemmata is a *lexicon.*

Example 5 (Finnish VH Lexicon)**.** *In Finnish the lexicon does not contain vowel delimiters, if we exclude compounds and build them in language production instead, since the vowel harmony is strictly covering a semantic entity and possibly more:* $\mathsf{fixVHECC} : \utilde{\Gamma}^*_\diamond \to P(\Gamma^*)$.

Example 6 (Turkish VH Lexicon)**.** *The lemmata mostly comply with the harmony* $\mathsf{tr}\mathscr{H}'$ *but having delimiters* $\natural$ *inside for the nested harmony during suffix harmonization. An examples is for a lemma not complying to* $\mathsf{tr}\mathscr{H}'$ *is* $\mathsf{el{\diamond}ma}$ *(apple). For Turkish the lexicon thus comprises words from* $\Gamma^*_{\diamond\natural}$.



(a) The nondeterministic transducer for transforming abstract utterances, thus without the pronounceable vowels from H_1 and H_2, into pronouncable utterances obeying the Finnish vowel harmony. All abstract vowels are nondeterministically replaced by their pronounceable counterparts.

(b) The deterministic transducer for the real-time vowel harmony error-correcting code which maps all remaining abstract vowels in the suffixes to pronounceable ones. This happens during speaking Finnish, e.g. *kivillä – on the stone*. It rejects utterances where a vowel harmony word does not comply with VH.

Figure 2: Transducers for utterances obeying the Finnish vowel harmony
The notation $H|H_i$ denotes the replacement of the abstract vowel a with its pronounceable vowel (a, i).

Both Turkish and Finnish have a finite list of suffixes in their lexica. Suffixes are the only lemmata of the lexicon which comprise the abstract vowels. One example are the question particles m$\textcircled{\tiny i ı u ü}$ and k$\textcircled{\tiny ö}$. Turkish can form a question by appending -mi, -mı, -mu and -mü, Finnish with -ko or -kö.

5.2 Real-Time VHECC

The VHECC which is conducted during the real production of NL, while speaking or writing, so to say in real-time, harmonizes those vowels which remained abstract after lexicon construction. Those vowels have to be mapped just-in-time. This imposes a high complexity to the speaker who has to formulate the semantics and in parallel respect all grammar phenomena including VH.

Definition 7 (Real-Time VHECC)**.** The *real-time* vowel harmony error-correcting code is the VHECC

$$\mathsf{tVHECC} : \Gamma_\diamond^* \to \mathsf{VHL}(\Sigma_\diamond)$$

that maps all abstract vowel $a \in H$ to their corresponding vowel $v = (a, i)$ for $i \in \tilde{h}$.

Example 7 (Finnish real-time VHECC)**.** *Finnish maps suffixes to its second harmony if a lemma comprises only neutral vowels.* Kivillä – on the stone *can be derived via the transducer in Figure 2b.*

Example 8 (Turkish real-time VHECC)**.** *Turkish exhibits no neutral vowels. Every assignment of a suffix to its harmony is clear-cut.* Mutluyum *means* I am happy *with the* tVHECC*-suffix* -yum *for* I am.

The real-time VHECC is the first one which does not accept all input. This would not be acceptable for technical ECCs as we noted already in §4 on page 4.

6 Capabilities of VHECCs

ECCs can detect and also correct corrupted code words. In this section we look at the capabilities of VH to detect and correct a NL utterance which is called a code word from coding theory's perspective.

Corruption can befall across a noisy channel. A noisy football stadium, a phone call, or just the NL conversation itself are examples for a noisy channel. Surprisingly, corruption is used on purpose, too. Then however it is not called like that but called a lossy compression. Corruption happens here for the sake of efficiency. We look at both unintended as well as intended corruption, i.e. compression.

6.1 Robustness by xVHECC

This subsection provides empirical evidence for our hypothesis that VH adds to the robustness of its language. Rephrased this means that we provide empirical evidence that VHECC is an ECC as defined in Def. 4. More specifically, we show this for the lexical VHECC (recall Subsection 5.1). By the analysis of this xVHECC in the Finnish dictionary, which was discussed in Example 5, we expect our experiment to show that fixVHECC fullfills the requirements of an ECC.

Minimal pairs are a powerful means to identify phonemes. Minimal pairs with respect to VH are therefore a suitable candidate for an experiment to shed light on whether VH acts as an ECC.

Definition 8 (VH Minimal Pair)**.** Two words $c_i \in$

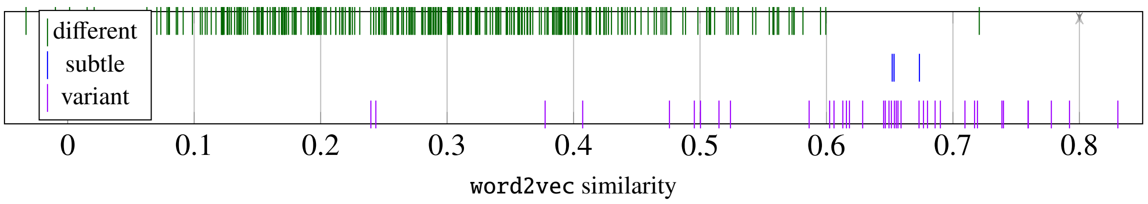


Figure 3: The `word2vec` embeddings show that most of the preVH homonym pairs convey two meanings which are quite different *(first line / green)* since the `word2vec` similarity value is low. Sometimes xVHECC seems to convey subtle meaning variants *(middle line / blue)*. The `word2vec` embeddings for preVH pairs with high similarity (> 0.6) are mostly mere variants with identical semantics *(last line / violet)*.
Thus, the experiment supports the thesis that primarily semantic distant homonym pairs are represented by the same string before applying xVHECC. This means that only semantic far away word pairs differ only with respect to VH. Similar word pairs strive for more difference than just chossing different VH classes.

Σ_i^* and $c_k \in \Sigma_k^*$ with $c_i \neq c_k$ are called a *VH minimal pair* if they are derived from the same raw word w, thus $c_i, c_k \in \mathsf{VHECC}(w)$. We call both c_i and c_k a *preVH homonym*.

For Finnish, we performed an experiment on preVH homonyms. Figure 3 shows the `word2vec` values for all preVH. On manual classification of the homonyms, we, as native speakers, identified three clearly distinct types of minimal pairs. We were able to classify every pair doubtlessly. Color coded with green, blue and violet, the plot shows semantically distant pairs, subtle meaning nuances and mere spelling variants of the same meaning, respectively from top to bottom. Roughly between 0 – 0.6, the semantically uncorrelated pairs *(first line / green)* lie – obvious homonyms in the classical sense namely same surface form with far away meanings. These lemmata convey quite different semantics. Rarely scattered in this segment, too, but mostly collapsed together in the adjacent segment of 0.6 – 0.9, the spelling variants *(last line / violet)* appear, such as `tagi` – `tägi` (*engl.: tag*). At the beginning of this segment the subtle meaning variations *(middle line / blue)* are scattered. For a handful of pairs the VH variants constitute a subtle meaning difference, such as `pörinä` – `porina`. Those are both onomatopoeiae but connotate slightly different nuances of the same meaning. In this category, VH serves as fine-grained semantic nuance and not as an ECC. While the majority of the results in the `word2vec` space divides neatly into expected low and high distance range, there are a few outliers in the results. We take this to be expected noise in the data, i.e. some of the meanings have not received correct values in the `word2vec` space. The only outlier for the different meanings is `taki` – `täki` (*engl.: coat – quilt*). The spelling variants *(last line / violet)* scattered in the segment 0 – 0.6 of the different meanings *(first line / green)*, but also spelling variants in general seemed to have not been identified by the `word2vec` calculations correctly as meaning exactly the same. The `word2vec` values indicate correctly their high similarity but should be higher to show the equality.

We interpret the experiment as follows: Technical ECC design strives to maximize the distance between all code words. Seldom, VH seems to transmit semantic nuances, splitting one meaning into a pair with slightly different but roughly the same meaning. Here, VH transmits semantic information. For most pairs in the experiment however, VH disambiguates semantically distant pairs with zero distance of the raw words – prior to VHECC, see again Def. 8. These different meanings can often be disambiguated by the context. If so, VH acts here as an ECC, contributing to the robustness. The Levenshtein metric is a suitable metric for VH words but not for preVH homonyms. Technical ECCs do not use semantic information of their to be processed data, see §5 on page 5. We can define a metric Δ that increments this Levenshtein distance by semantic distance between the lemmata in the VH word in some suitable canonical way. One way would be to project the `word2vec` similarity to a distance value in $[0 - 0.5]$ in order to retain the triangle inequality required for a proper metric.

A speaker says $u \in \Sigma_\diamond^*$ where a VH minimal pair occurs: $c = c' = \mathsf{ECC}(u)$ with w as a substring of u. The information i answers the question: which one of the two meanings of the occurring homonym was meant? The speaker has u with w as a substring in mind – w one of the two meanings of the homonym w'. Speaking does not transmit the meaning of w but only the pure u which includes w'. Consequently, we view c already as corrupted: $c' = c$. As corruption $'$ we thus use the deletion of the unambiguous meaning. w' is the pair of preVH homonyms, one of them is w. Obviously the formula in Def. 4 is met, since $i_{c'}(c') = i$ can disambiguate the meaning due to VHECC. $\Delta\,(i_{c'}(c'), i_w(w))$ is zero for all w – the information can be fully retrieved. $\Delta\,(i_{w'}(w'), i_w(w)) \geq 0$ since we can either disambiguate the homonym's meaning from the context, then $i_{w'}(w') = i$, or we cannot. Note that Δ must operate not only on strings but also on meanings defined in some canonical way. We conclude that VHECC for utterances including a preVH homonym with respect to the corruption being the noiseless transmission and asking for disambiguation of the homonyms meets the definition of an ECC.

6.2 Lossy Decompression via VHECC

Error-correcting facilities may act as a decompression. A lossy compression is a corruption of the input which is done on purpose. Viewed that way, the lossy compression corresponds to the noisy channel. If this input was enriched via an ECC, thus before being compressed, the decompression can be conducted with the help of this ECC.

In spoken language we find long utterances without a break. Until there is a silence with no voice, it takes several written words. Delimiters, like written spaces, are compressed away. We could not speak fluently and fast enough if we would stop our voice after every semantic entity or at every VH delimiter. Compressing VH delimiters is done on purpose. VHECC addresses this lost tokenization of the input stream in language perception. The VHECC decoding is a decompression. Note that the compression of delimiters may be lossy meaning that we cannot reconstruct all of the delimiters. In particular, reconstruction of the exact positions of delimiters cannot be guaranteed. If words of the same harmony ensue each other we lose the delimiter completely. If two words of different harmonies are adjacent, we know the exact position of the delimiter only if the vowels of the distinct harmonies which are in juxtaposition with each other.

Theorem 2 (Reconstruction of the Number of VH Delimiters)**.** Let u' be a nonempty error-corrected and corrupted utterance $u' \in \mathsf{cVHECC}'(u_{\mathsf{raw}})$ for all $u_{\mathsf{raw}} \in \underset{\sim}{\Gamma}_{\diamond}$. The channel $'$ is the loss of VH delimiters resulting in $u' \in \Sigma^*$. As information i we are able provide the minimum number of reconstructable VH delimiters as well as the maximum number. VHECC is an ECC with respect to the noise type of VH delimiter deletion and the information i.

Proof. The *minimum number* of VH delimiters in the raw utterance equals $|\tilde{h}|$, since we can group all words of one harmony H_i next to each other. Only adjacent words from different harmonies reveal a vowel harmony delimiter. In an uncompressed utterance every word's end is indicated by the VH delimiter. The decompression thus reconstructs $|\tilde{h}|$ as the minimum number of delimiters, since in above arrangement of the words we can reconstruct only that a harmony ends, not that a word ends.

Let $\#i$ be the number of words w in a harmony H_i thus $w \in \underset{\sim}{\Sigma}_i^*$ with w includes at least one vowel from H_i. To reach the maximum number of reconstructable spaces, we alternate and iterate through the harmonies. Alternating between two types of words is sufficient to make a delimiter visible. We denote words of the harmony which occurs most often[4] in u with x. The words y are drawn from all the other harmonies. We denote the cardinality of those two sets of words by $\#x$ and $\#y$, respectively:

$$\#x = \max_{i \in \tilde{h}} \#i \quad \text{and} \quad \#y = \left(\sum_{i \in \tilde{h}} \#i \right) - \#x.$$

We lose only VH delimiters when we cannot alternate anymore which is the case if $\#x > \#y + 1$. The harmony with the most words in u, denoted by H_x, does show whether delimiters are not reconstructable, since in that case H_x has no counterpart to alternate with. Thus from the words y we take aside an amount of $\#x - 1$ for separating all x from each other. In every alternation step we additionally iterate through the harmonies to place also the remaining words y. The *maximum number* of VH delimiters is either $(\#x - \#y - 1)$ or 1.

Now we proof that cVHECC is an ECC. This is now obvious for both the information i of the maximum or the minimum number of $\diamond$s or even the combination of those by subtracting the minimum from the maximum number. Since the corrupted, not error-corrected utterance u'_{raw} has no information about the delimiters except the end of the utterance for every u'_{raw} both the maximum as well as the minimum number of reconstructable $\diamond$s is one. This is less than the i for u' or the same.

[4] The x stands for the x in maximum.

Consequently cVHECC guarantees that it is advantageous since this means that the sum in Def. 4 is always greater than zero. □

7 Conclusion

This paper formalizes VH from a viewpoint of formal language theory (FLT) and automata theory. This viewpoint suggests itself since FLT and the corresponding automata are the traditional approach in literature to formalize NLs' theoretic aspects. This formalism allows us to compare vowel harmony to ECCs defined originally for technical applications only. We could show that our hypothesis bears this comparison with empirical and formal evidence. (1) PreVH homonyms, lemmata which would be homonyms without VH, exhibit mostly a high semantic difference. This shows that VH is used to enlarge the distance between lemmata which in turns shows that VH acts as an ECC. (2) The compression via delimiter elimination conducted in spoken language is lossy but vowel harmony delimiters can partly be found via error-correction. In turn, those delimiters help to reconstruct the original tokenization. In conclusion, we can state that VH indeed serves as a mechanism for error-correction in communication.

8 Acknowledgements

Teodora Nasz discussed the generalization of the error correcting code definition with me when I struggled with the right level of abstraction. Timofey Arkhangelskiy, Andreas Maletti and Michael Wehar provided valuable feedback on several drafts. Usama Sadar helped by his feedback on the caveats of how to present interdisciplinary results. Last but not least I thank all anonymous reviewers.

References

Anatoly V. Anisimov and Igor O. Zavadskyi. 2014. Forward error correcting codes by means of the two-base (2, 3)-numeration system. In *BlackSeaCom*, pages 107–111. IEEE.

Diana Archangeli and Douglas Pulleyblank. 2007. *Harmony*, Cambridge Handbooks in Language and Linguistics, page 353–378. Cambridge University Press.

Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity - A Modern Approach*. Cambridge University Press.

Dinah Baer-Henney, Frank Kügler, and Ruben van de Vijver. 2015. The interaction of language-specific and universal factors during the acquisition of morphophonemic alternations with exceptions. *Cognitive Science*, 39(7):1537–1569.

Elwyn R. Berlekamp. 1968. *Algebraic coding theory*. McGraw-Hill series in systems science. McGraw-Hill.

Arthur Brady, Kyle Maxwell, Noah Daniels, and Lenore J. Cowen. 2009. Fault tolerance in protein interaction networks: Stable bipartite subgraphs and redundant pathways. *PLoS ONE*, 4(4):e5364.

Elena Dubrova, Mats Näslund, Göran Selander, and Karl Norrman. 2016. Error-correcting message authentication for 5g. In *MobiMedia*, pages 149–158. ACM.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, pages 381–390. ACM.

Sara Finley and William Badecker. 2008. Analytic biases for vowel harmony languages. *Proceedings of the West Coast Conference on Formal Linguistics*, 27:168–176.

Richard Wesley Hamming. 1950. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In *MOL*, pages 52–63. ACL.

Nicholas Henriksen. 2017. Patterns of vowel laxing and harmony in iberian spanish: Data from production and perception. *Journal of Phonetics*, 63:106–126.

Hyun Jin Hwangbo. 2015. Learnability of two vowel harmony patterns with neutral vowels.

Wendell Kimper. 2017. Not crazy after all these years? perceptual grounding for long-distance vowel harmony. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 8(1).

Martin Krämer. 2003. *Vowel Harmony and Correspondence Theory*. DE GRUYTER.

Maria-Rosa Lloret. 2008. On the nature of vowel harmony: Spreading with a purpose. *Proceedings of the XXXIII Incontro di Grammatica Generativa (Bologna*, pages 15–35.

Juhani Luotolahti, Jenna Kanerva, Veronika Laippala, Sampo Pyysalo, and Filip Ginter. 2015. Towards universal web parsebanks. In *Proceedings of the International Conference on Dependency Linguistics (Depling'15)*, pages 211–220. Uppsala University.

Bashar M. Mansoor and Tarik Z. Ismaeel. 2019. Design and implementation of an improved error correcting code for 5g communication system. *JCM*, 14(2):88–96.

Oleksandr Marchenko, Anatoly Anisimov, Igor O. Zavadskyi, and Egor Melnikov. 2018. English text parsing by means of error correcting automaton. In *NLDB*, volume 10859 of *Lecture Notes in Computer Science*, pages 281–289. Springer.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2015. Computing numeric representations of words in a high-dimensional space. Patent, Publication Number 09740680. Google Inc.

Elliott Moreton and Joe Pater. 2012. Structure and substance in artificial-phonology learning, part ii: Substance. *Language and Linguistics Compass*, 6(11):702–718.

Saleh M. Mwachaka, Aiping Wu, and Qingqing Fu. 2019. A review of mud pulse telemetry signal impairments modeling and suppression methods. *Journal of Petroleum Exploration and Production Technology*, 9(1):779–792.

Tommi A Pirinen. 2015. Development and use of computational morphology of Finnish in the open source and open science era: Notes on experiences with omorfi development. *SKY Journal of Linguistics*, 28.

Anne Pycha, Pawel Nowak, and Eurie Shin. 2003. Phonological rule-learning and its implications for a theory of vowel harmony. *Wccfl*, 22:423–435.

Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. 1999. Pseudorandom generators without the XOR lemma (extended abstract). In *STOC*, pages 537–546. ACM.

TurkuNLP. 2019. Finnish embeddings. 2019-03-13 `http://dl.turkunlp.org/finnish-embeddings`.

Felix Winter, Nysret Musliu, and Peter J. Stuckey. 2020. Explaining propagators for string edit distance constraints. In *AAAI*, pages 1676–1683. AAAI Press.

Johannes Van Wonterghem, Amira Alloum, Joseph Jean Boutros, and Marc Moeneclaey. 2018. On short-length error-correcting codes for 5g-nr. *Ad Hoc Networks*, 79:53–62.

Igor O. Zavadskyi. 2015. Variable-length error-correcting codes based on finite automata. *Cybernetics and Systems Analysis*, 51(2):198–204.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task, EMNLP'18*.

Yihan Zhang, Amitalok J. Budkuley, and Sidharth Jaggi. 2020. Generalized list decoding. In *ITCS*, volume 151 of *LIPIcs*, pages 51:1–51:83. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.