MUSICOLOGIST
International Journal of Music Studies
Trabzon University State Conservatory © 2017-2025
Volume **9** Issue **1** **June 2025**

Research Article
Musicologist 2025. 9 (1): 1-29
DOI: 10.33906/musicologist.1556608

FRANK SCHERBAUM
Potsdam University, Germany
fs@geo.uni-potsdam.de
orcid.org/0000-0002-5050-7331

SIMHA AROM
Paris Diderot University, France
simha.arom@mnhn.fr
orcid.org/0000-0002-2129-9190

FLORENT CARON DARRAS
PSL University, France
florentcarondarras@gmail.com
orcid.org/0000-0003-3869-1399

What can we learn about the grammar of traditional Georgian vocal music from computational score analysis?

ABSTRACT
This paper describes the current status of a long-term project aimed at understanding the chordal syntax of traditional Georgian vocal music by analyzing sheet music in Western 5-line staff notation. As an important milestone, we present a generative grammar model based on the self-learning Kohonen model (Kohonen, 1989) in a prefix tree (Antonov, 2018; 2023) framework. This represents a significant improvement over the classical Markov model, as it allows for the influence of different context lengths for each chord in a chord sequence. We used this model to generate a large number of chord sequences, all conforming to the same grammatical production rules as our corpus. These were then used as training data for an artificial neural network to test whether, as in large language models (LLMs), 'linguistic relationships' could be identified by visually analyzing the embedding space of the network. The results for chord-to-chord relationships are inconclusive, as the spatial structure of the embedding map for individual chords cannot be interpreted unambigously. The embedding map for whole songs, however, shows a pronounced spatial clustering which reflects the different classes of our corpus. This suggests that the structure of the embedding map reflects the similarities and dissimilarities of the chordal syntax of the individual songs, which the network has learned in an unsupervised way.

KEYWORDS
Traditional Georgian Vocal Music
Musical Grammar
Machine Learning
Kohonen grammar

Received: September 26, 2024; **Accepted:** March 30, 2025

Introduction

The country of Georgia, located at the crossroads of Europe and Asia, has an incredibly rich heritage of musical traditions that have been passed down for centuries through oral tradition. In this context, regular communal singing, which is still actively practiced, at least in part, in some rural areas of the country, has proven to be a crucial mechanism. It has preserved the transmission of knowledge between generations and helped maintain the vitality of this music. While — through many years of practice and continuous exposure — this mechanism has led to the development of an intuitive understanding of the music's grammar (tonal organisation, musical syntax, and other important elements), this knowledge, as long as it is not formalized, represents a cultural heritage that remains immaterial in the truest sense of the word. However, what does exist in material form are numerous transcriptions of performances in the form of musical scores in Western 5-line staff notation.

The focus of our collaboration, which began in 2014, is the question of how knowledge about the chordal syntax of traditional Georgian vocal music can be gained — at least in part — from such transcriptions, with the help of tools from computational ethnomusicology, computational linguistics, machine learning, and artificial intelligence (AI) research.

Our work, which can be seen as a follow-up study to the papers by Arom and Vallejo (2008; 2010), raised a large number of questions along the way, for which there was no clear answer in terms of an existing "best practice". For example:

How should one compare songs notated in different keys? Should all songs be transposed to the same key prior to analysis, or not? Should one work with a) absolute pitches, b) relative pitches, or c) scale degrees relative to a chosen reference note? In the case of c), what reference note should be chosen? Should scores be reduced to their presumed harmonic pillars prior to analysis, or not? If so, would that not bias the results by adding too much subjectivity? If not, how could one manage the enormous amount of information? Should one work with the original scores in Western notation at all, or should they be transformed into a more unbiased representation?

At times, ad hoc choices needed to be made, some of which led to conceptual dead ends,

requiring a time-consuming rethinking of the entire processing chain. Based on the results of acoustical studies on the tonal organization of traditional Georgian music (Scherbaum, et al., 2020; Scherbaum et al., 2022; Tsereteli and Veshapidze, 2014; 2015), we realized that analyzing traditional Georgian music in terms of Western church modes, as attempted by Arom and Vallejo (2008), could result in artifacts that were distorted by the transcription process and needed to be abandoned. In order to at least partially correct the available scores (which are all notated in 12-TET tuning in different keys) for the distortions caused by the transcription process, we finally adopted the procedure of Scherbaum et al. (2024). It is based on first transforming the original pitches into scale degree indices (SDI) with respect to the notated key, which can then subsequently be transformed into any heptatonic tuning system. In this context, the SDI notation represents the pitches in a manner that remains independent of the details of the actual tuning system, as long it is heptatonic. For more details see Scherbaum et al. (2024).

Despite these challenges – and even during the pandemic when we could only communicate remotely – we managed to continue our collaboration and make progress. For example, a comparative study of a small collection of Georgian and Medieval polyphonic songs (Arom et al., 2018) taught us that it is mainly the chordal syntax and not the chord inventory that structurally distinguishes the two collections. This gave some direction to our research. By the time the Covid restrictions ended, and we could finally meet in person again, we had significantly expanded our digital corpus to more than 450 songs. This turned out to be a large enough cohort to demonstrate that different regional and stylistic subsets in the corpus could be computationally distinguished through differences in their chord-progression patterns (Scherbaum et al., 2024).

Since the public release of ChatGPT in November 2022, it has become clear that new developments in the field of natural language processing have much to offer for the questions we seek to answer within our collaboration. As a result, we have started to explore how to utilize these developments in our analysis.

In the following sections, we will discuss what we believe can be learned about the grammar, in particular the chordal syntax, of Georgian traditional vocal music by analyzing musical scores using a variety of tools and approaches available today. Rather than simply presenting the results for our current corpus, the details of which will evolve

as the corpus grows, we aim to focus on the discussion of the main concepts underlying our perspective.

Overall, we view this work as a feasibility study with the long-term goal of developing building blocks for an optimal workflow to decode and better understand the rules underlying the chordal syntax of traditional Georgian vocal music.

Methodological Considerations

Understanding the grammar of a language or musical system requires knowledge of its intrinsic structural patterns. Only when we know these can we form intelligible and grammatically correct sentences. For the following considerations, we define — following linguistic practice — the search for the grammar of Georgian traditional music as the search for the set of rules that describe the construction and structure of this music. In the context of our project, in which the focus is on chordal syntax, we have explored several perspectives for representing songs, each of which has different advantages and disadvantages.

Songs as sequences of images

It is well known that humans are generally very skilled at recognizing visual structures. Sometimes, this ability is so powerful that it leads to the phenomenon of pareidolia[1]—the erroneous assignment of familiar patterns, such as faces, to diffuse perceived structures, like clouds. To apply this sensitivity to visual pattern recognition to our task, one of the approaches we pursued was converting musical chords into images, which were then analyzed visually. The challenge in this context was finding forms of visualization that did not significantly reduce the information content of a score while remaining easily perceivable in their entirety. Not all of these representations were convincing in terms of their applicability to collections of songs, but most proved to be quite useful for the structural analysis of individual scores. Moreover, many of these visualizations were aesthetically fascinating in their own right, including Harmonygrams (Scherbaum, 2024; Scherbaum et al., 2024; Scherbaum and Mzhavanadze, 2024).

As an example, Fig. 1 shows a Harmonygram for the chant *Dghres Saghvtoman Madlman*, in which the individual phrases are arranged vertically. The melodies of the three voices

[1] https://en.wikipedia.org/wiki/Pareidolia

are displayed in Global Notation (Killick, 2021), while the colored interval columns indicate —from top to bottom— the harmonic intervals between the top and middle voices, the middle and bass voices, and the bass and top voices, respectively. Each chord is associated with a unique colored pattern. The colored rounded rectangles mark segments of the chant that are identical (marked by identical colors) or similar (marked by similar colors, e.g., green) to help identify related chord progression sequences.

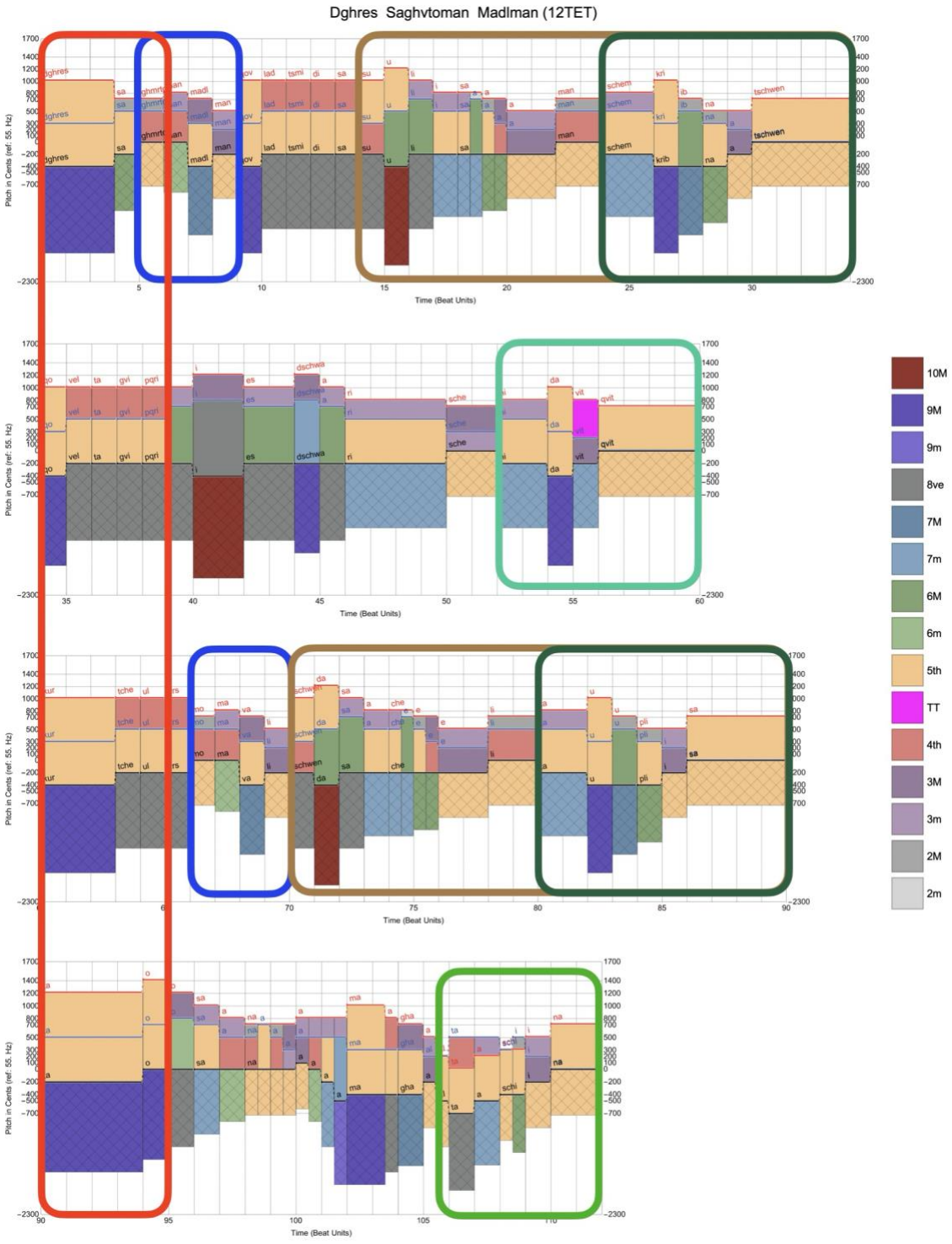

Figure 1. Harmonygram of the chant *Dghres Saghvtoman Madlman*. For detailed explanation see text and **(**Scherbaum, 2024**;** Scherbaum et al., 2024**;** Scherbaum and Mzhavanadze, 2024**).**

The only graphical approach for structural analysis that holds essentially the same power for analyzing individual songs as it does for an entire corpus is the representation of

songs as directed graphs (or 'song paths' on a 'chordscape')[2] (Scherbaum et al., 2016a; 2016b). Graphs effectively visualize the temporal structure of chord progression sequences, while simultaneously allowing for mathematical analysis. In other words, they can be used as mathematical objects that can be manipulated algorithmically. For example, one can calculate distances between graphs, which, in turn, allows for the quantitative exploration of neighborhood relations and similarities between the underlying sequences. Analyzing songs as directed graphs allowed us to demonstrate that the differences between medieval and Georgian songs are primarily differences in the syntax of chord progressions rather than in the chord distributions themselves (Arom et al., 2018).

One of the open problems with the 'graph approach' is that a musically meaningful solution to the graph layout problem has yet to be found. This is the main reason we have not yet applied it to the analysis of our entire corpus.

Songs as sequences of string tokens

For the analysis of our entire corpus and its subsets, we exploited the fact that a song can also be written as a sequence of so-called string tokens, which, for our purposes, can be treated like words in a natural language. For example, a chord represented by the notes C4, E4, and G4 (in scientific pitch notation[3]) and a duration of 2.5 quarter notes could be expressed as the string token 'C4_E4_G4_(2.5)' and then converted back to its original representation without losing any information. As a result, a whole song, expressed as a sequence of string tokens, can be computationally processed like a sentence in an unknown language. Just as text fragments contain information about the grammar of the language in which they were written, we assume that musical scores contain information about the grammar of the music encoded within them, and that this information can be extracted in a similar way.

For subsequent analysis, all the scores were transformed into sequences of string tokens so that they could be modeled using the same tools that are used for natural language processing —such as those used to produce so-called language models. Language models

[2]https://www.uni-potsdam.de/de/soundscapelab/computational-ethnomusicology/scores-chordscapes-and-song-paths

[3] An alternative which is better suited for algorithmic processing is the representation of the note pitches in scale degree index (SDI) notation described in detail in (Scherbaum et al., 2024).

are generative probabilistic models that predict the next word in a sentence based on the preceding words. What is currently referred to as large language models (LLMs) are special types of artificial neural networks (ANNs), trained on vast amounts of text, such as books, articles, and websites. One remarkable feature of these models is their apparent deep understanding of how language works, including grammar, vocabulary, and various topics. With massive training data (for instance, OpenAI's GPT-3 model was trained on hundreds of gigabytes of text), these LLMs can generate human-like responses that seem coherent and contextually appropriate.

One of the questions we pursued in our work was to what degree we could benefit from these modern developments, despite the fact that our datasets are tiny in comparison to the vast amounts of data typically used to train LLMs, making direct training of an LLM from our data unfeasible. However, other types of statistical models, such as n-gram models or the Kohonen model (Kohonen, 1989), though lacking some of the power of LLMs, turned out to offer significant advantages while being far less demanding in terms of data size.

For this study, these models were implemented using a special data structure called prefix trees, or tries (Antonov, 2018; 2023). The trie representation is known to be extremely efficient for tasks like automatic word completion during text input in word processing programs. In our case, when combined with the Kohonen model, it proved useful for two different purposes: first, as part of a generative model for producing synthetically generated, grammatically correct new songs; and second, for the analysis and visualization of the rule set of the detected Kohonen grammar, as will be described below.

From Markov to Kohonen

Kohonen's self-learning musical grammar model (Kohonen, 1989) can be seen as an extension of Markov chain models, also known as n-gram models (Scherbaum et al., 2024; Scherbaum et al., 2015). Markov models operate on the so-called 'Markov assumption,' which states that the probability of a future state in a sequence (e.g., a chord in a song) depends on the current state and —depending on the order of the Markov process— a few predecessor states. In other words, in the context of language modeling, n-gram models predict the next word based on the current word and the n-1 prior words. The

probability of a particular transition can be derived from a set of example data by a simple bookkeeping exercise on their n-gram set. Kohonen's self-learning grammar model (Kohonen, 1989), on the other hand, goes beyond plain Markov models by dynamically expanding its rule-set context based on the input it receives, allowing it to capture some long-range dependencies individually (based on what Kohonen called 'logical conflicts' in the production rules (Kohonen, 1989)).

To explain the principles, let's assume we have a sequence of 'states'—for now represented by different letters—that have been produced according to rules unknown to us. From the Markov model perspective, one assumes that the occurrence of a particular state at position k in the sequence depends on a) the particular state and b) possibly what has happened prior to position k.

In the unigram (1-gram) model, it is assumed that it does not matter at all what has happened prior to position k and that the probability of occurrence of a particular 'state' – let's say state X – depends only on the overall 'probability' of state X occurring. Prob(seq[[k]] equals X) = Prob(X).

In the bigram (2-gram) model, it is assumed that the probability of occurrence of a particular state at position k in the sequence depends on the previous state of the sequence seq[[k-1]] at position k-1, and the transition probability from the previous state of the sequence to X. One can also say that in the bigram model, one considers a context of length 1 for the state of interest.

The trigram (3-gram) model follows the same construction principle, except that the context length is extended to 2. Finally, in the general case of an n-gram model, one considers a context length of n-1. In other words, one assumes that each state in a sequence has been influenced by n-1 previous states. The length of the considered context, namely the value of n-1, is also referred to as the order of the Markov process.

All the constituents of the n-gram models can be calculated by simple bookkeeping exercises from 'training data,' which represent a particular set of sequences, e.g., words, notes, or chord sequences. Despite its simplicity, n-gram models have been quite successful in music analysis, such as for the classification of musical scores (Scherbaum et al., 2024).

However, n-gram models are missing something very important, particularly for the analysis of music, by making the assumption that the length of the relevant prior context is fixed. This is rather unrealistic. Some chords in a musical chord sequence may reasonably be modeled as being influenced only by the previous chord, while others may have been influenced by two, three, or more prior chords. Therefore, it is much more realistic to assume that the length of the relevant prior context of a particular state will depend on the state itself.

This is where the Kohonen model performs much better, because it does not assume a fixed-length context. Instead, within the Kohonen framework, it is assumed that the number of predecessor states on which the 'next state' depends —referred to as the 'relevant context' of the current state— may vary. Similar to the n-gram model, all the necessary ingredients of the Kohonen model can be determined (learned) from training data. The lengths of the relevant contexts for a particular training data set are determined dynamically based on logical conflicts that occur in the training examples (Kohonen, 1989).

To see what this means in practice, let's look at the example sequence of states from Kohonen's paper, which is shown in Fig. 2a.

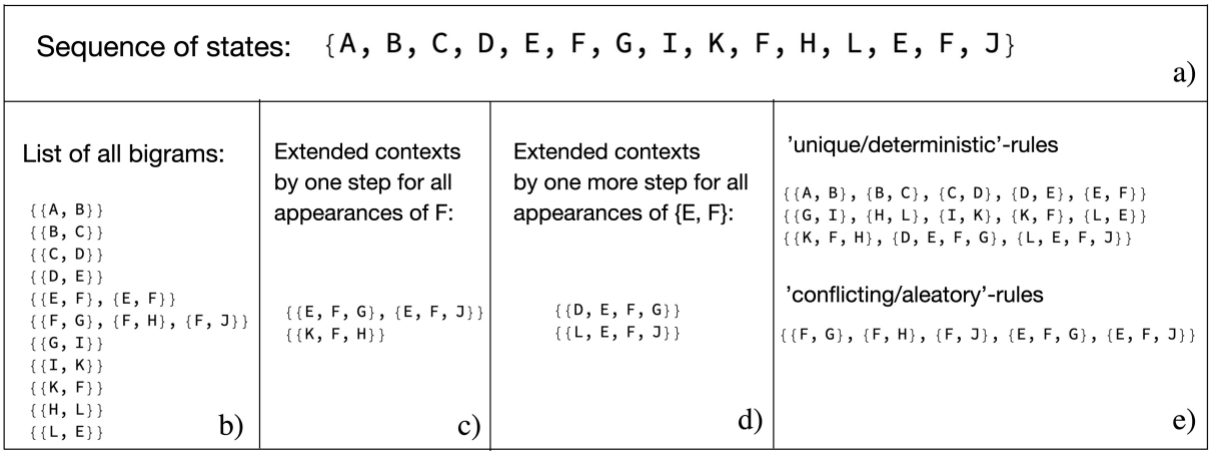

Figure 2. The determination of the production rules for a sequence of states in the Kohonen model.

Fig. 2b) shows the list of all the bigrams in this sequence, sorted according to their first element. One can immediately see that there is a logical conflict in the transition from the letter F because F goes to G once, to H once, and to J once. All the other letters are always

followed by the same next letter. The transition from E to F appears twice, which is not a logical problem; on the contrary, it could suggest this being a strong rule.

Therefore, F is what Kohonen would refer to as a 'conflict' case. To resolve this, Kohonen dynamically extends the contexts for all instances of the letter F by one. The result is shown in Fig. 2c). You can see that one of the conflicts is resolved this way: If F is preceded by the letter K, the next state is H, and the conflict is resolved! However, this does not help when F is preceded by the letter E. In this case, the next state in the token sequence is G once and J once. Thus, in this case, the context still needs to be extended one more step further. The result is shown in Fig. 2d).

Now, all conflicts are resolved. If F is preceded by an E which is preceded by a D, the next letter is G, while if F is preceded by an E, which is preceded by an L, the next letter is J.

As a result, instead of representing a sequence of tokens by a sequence of sub-sequences of fixed lengths (i.e., a sequence of n-grams), Kohonen's approach leads to the representation of the input sequence by a sequence of sub-sequences of variable lengths, representing unique 'production rules'. These rules, which for this example are shown in the top panel of Fig. 2e), can also be seen as deterministic production rules or 'always' rules, because their contexts will always lead to the same 'next state'.

However, the contexts that still contain conflicts are also important structural elements! They represent what one could call 'sometimes' rules, or aleatory rules, and are shown for our example in the bottom panel of Fig. 2e). These rules indicate that a particular sub-sequence is sometimes followed by, for example, state X, and sometimes by state Y. If we count the number of times each of the 'sometimes' states occurs, we have all the information needed about the statistics of these aleatory rules.

In conclusion, the Kohonen model allows for the determination of a set of (deterministic and aleatory) production rules for a sequence of states. These rules can be encoded in a simple table, which is referred to as Kohonen's 'memory,' as shown in Fig. 3a).

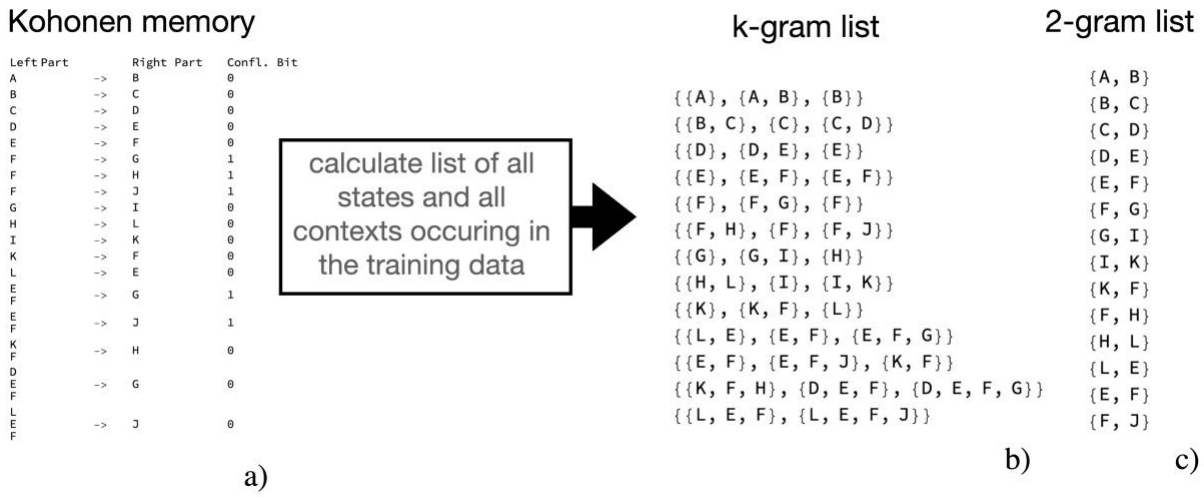

Figure 3. From the Kohonen memory table to the k-gram list.

Each line in the Kohonen memory table corresponds to a transition that is realized in the training data. The right column in Fig. 3a), labeled the 'conflict bit,' indicates whether the transition is part of an aleatory rule (conflict bit is ON) or a deterministic rule (conflict bit is OFF).

With the information now available, it is a simple bookkeeping exercise to calculate the list of all states and contexts occurring in the training data. If this is done in such a way that the elements of this list occur in the same proportion as in the training sequence, one obtains what we refer to as the k-gram list shown in Fig. 3b). In this context, the 'k' stands for Kohonen. The k-gram list can be seen as the repository for the 'building blocks of the syntax' that represent the rules of the learned grammar. At first glance, it looks like a mixture of 1-grams, 2-grams, 3-grams, and 4-grams. In Markov chain terms, the complete set of 1-grams generated from a sequence of tokens tells us how often a particular token is present in it, while the set of 2-grams represents the frequency-of-occurrence distribution of transitions from one token to another, and so forth. The k-gram list, however, combines subsets of n-grams of different orders. The selection occurs based on logical conflicts—in other words, situations where the transition from a particular state to the next is not uniquely defined unless the number of predecessor states included in the context is increased. The maximum length of a k-gram is determined by how many prior states the algorithm has to consider until the 'Kohonen memory' (see above), which is built up during the learning phase, no longer changes.

Comparing the k-gram list to the bigram list shown in Fig. 3c), one can immediately see how Kohonen's model extends the Markov chain model, leading to a much richer representation of the syntax of the training data.

Prefix trees

Applied to a corpus of songs, the Kohonen grammar model contains everything — assuming that only the immediate prior context is relevant— that can be determined from a corpus of scores about the syntactic structure of the music it represents. However, the way it is represented in the computer, either as a Kohonen memory table or as a k-gram list, results in a large table that requires additional tools for analysis and visualization. The solution to this challenge is a data structure called a prefix tree, or simply a trie (Antonov, 2018; 2023).

To illustrate the principle of constructing the prefix tree for our example, the various contexts in the k-gram list are first grouped according to their first elements and sorted vertically from bottom to top: A, B, etc., as shown in Fig. 4a).

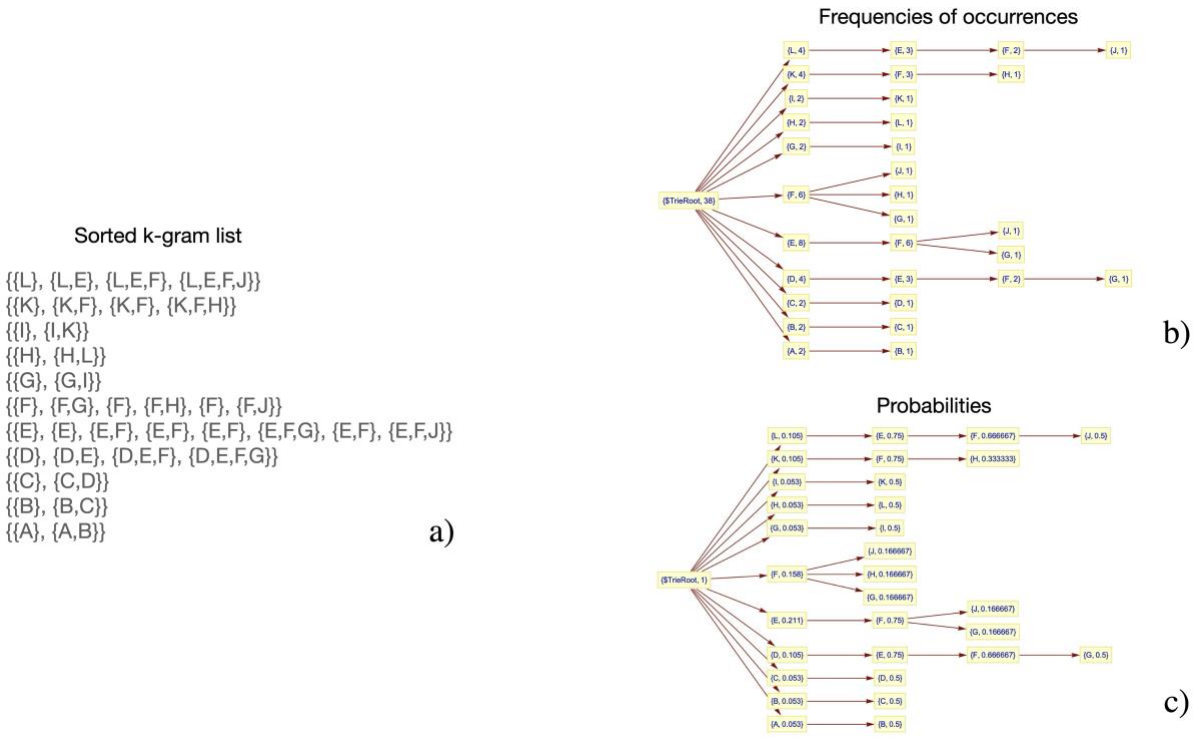

Figure 4 a) Sorted k-gram list, b) prefix tree for frequencies of occurrence, and c) for probabilities, respectively, are shown for the running example.

Fig. 4b) shows the corresponding prefix tree. The root node (labeled $Trieroot)

represents the head of the tree. The children of this node represent all unique starting letters of all determined context lists. The information in the i-th row of the sorted k-gram list is mapped to the i-th branch of the prefix tree. The numerical values associated with each node simply indicate the number of occurrences of each element in a subsequence. The information given for each node is easily extracted from the values in the curly brackets. For example, the sub-sequence in the topmost branch, which starts with L, contains 4 Ls, 3 Es, 2 Fs, and one J.

By dividing the number of occurrences of a particular node in each of the sub-sequences in Fig. 4b) by the number of occurrences of the node above it, we obtain the conditional probability of reaching that particular node from the node above. The corresponding prefix tree is shown in Fig. 4c). As a result, for each node, we can now immediately calculate which letter could follow and with what probability. A common application of prefix trees is for the completion of word sequences typed into a mobile phone or word processing program, where the prefix tree has been trained with the complete vocabulary of the used language.

In our context, the prefix-tree structure has proven to be extremely efficient in two ways. First, as an engine of a generative model to produce synthetic, grammatically correct new scores (simply by randomly selecting root-to-leaf paths), and second, for the analysis and visualization of the rule set of the detected grammar, as illustrated in Fig. 5.

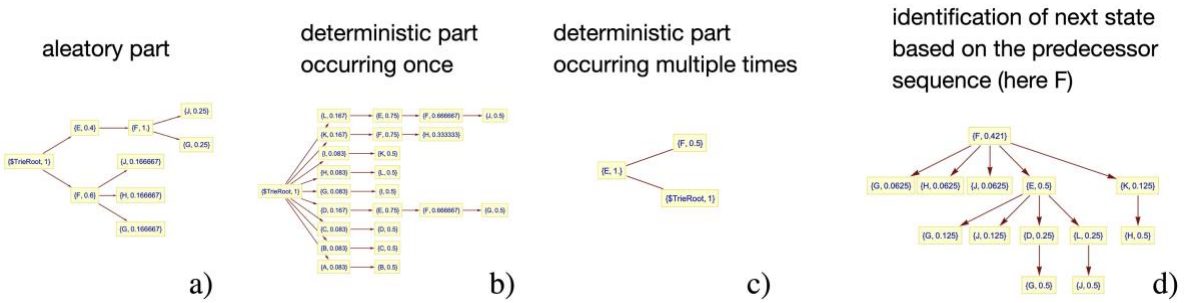

Figure 5. Exploring the rule set of the determined grammar according to different criteria.

It is now fairly easy to explore the tree structure according to certain criteria. For example, we can isolate the aleatory part, shown in Fig. 5a) ; the deterministic part occurring just once, shown in Fig. 5b) ; or the deterministic part occurring multiple times, as shown in Fig. 5c). Additionally, we can display the next state based on the predecessor

sequences, as shown in Fig. 5d).

Application: the Erkomaishvili dataset

In the following, we will discuss the application of the Kohonen model to a set of roughly 100 liturgical chants from the Shemokmedi Monastery in Western Georgia (Shugliashvili, 2014). This corpus is based on audio recordings of the master chanter Artem Erkomaishvili from 1966. For several reasons, this is currently our preferred study object for score-based corpus analysis. First, the transcriptions by David Shugliashvili are publicly available in digital form (Rosenzweig et al., 2020; Shugliashvili, 2014). Second, we have already used this dataset in several prior studies (Rosenzweig et al., 2020; Scherbaum et al., 2020; Scherbaum et al., , 2021; 2023) and are therefore familiar with some of its characteristics. Finally, the transcriptions by David Shugliashvili mark the individual phrases of each chant, which allows us to also use the full Harmonygram perspective (Scherbaum, 2024; Scherbaum et al., 2024; Scherbaum and Mzhavanadze, 2024) for each of the chants as an additional means of analysis.

The determination of the Kohonen grammar for this dataset results in a total of nearly 14,000 production rules. These can easily be stored in a prefix tree, but it is obvious that they cannot be analyzed simply by visual inspection. Even restricting ourselves to the deterministic rules that occur more than once does not solve this problem, as there are still more than 3,000 such rules. This is because, within the Kohonen model framework, every single chord in the corpus is modeled as the result of applying a production rule. Obviously, not all the rules are equally representative of the underlying grammar. Some of them, perhaps even the majority, may simply represent ornamental elements, which are only of secondary interest to us at this time.

Given this challenge, we felt it necessary to develop and explore various strategies for further action. For one, we are currently exploring to what degree ornamental elements of a chant can be removed from a score with the help of Harmonygram analysis. This is a very time-consuming, manual process for which we do not yet have final answers, as we have only recently started with this approach (Arom and Scherbaum, 2024).

Additionally, we have begun to investigate the Kohonen model of the Erkomaishvili corpus through what could be termed ‘specific questioning’, which is done in a way that

allows the answers to be computed with the help of the prefix tree. The questions we have considered specifically with respect to the Erkomaishvili corpus are, for example: What are the most often used production rules? What are typical cadences? What are the most likely chords to follow a particular chord? This line of inquiry leads directly to the problem of using the production rules as a generative model. In the following, we will consider these questions one by one.

What are the most often used production rules?

Figure 6 shows the "root-to-leaf-path representation" of the 30 most frequently used deterministic production rules in the Erkomaishvili corpus. The numerical values in the string tokens in Figure 6a are in scale degree index (SDI) notation, as described in Scherbaum et al (2024**)**

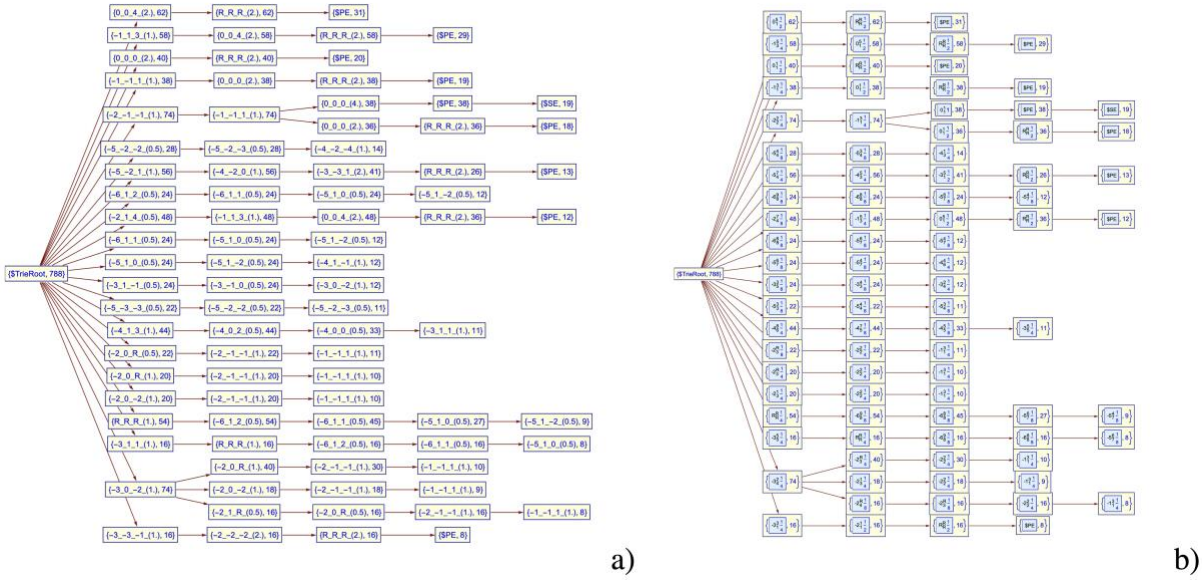

Figure 6. "Root-to-leaf-path representation" of the 30 most frequently used deterministic production rules in the Erkomaishvili corpus. The corresponding tree nodes are shown as string tokens and subscript-superscript chord symbols and in figure a) and b), respectively.

Although Figure 6 contains complete information about the chord progressions in each of the production rules, this form of visualization is difficult to perceive because one must read the information for each node sequentially and retain it in memory. Even with practice, this remains a "slow" process in the Kahneman sense (Kahneman, 2013). Perception becomes slightly faster if the nodes are written in subscript-superscript chord notation (Figure 6b), but the fundamental perceptual problem persists.

A much faster way to perceive the information in Figure 6, which also facilitates the comparison of individual production rules, is to display the nodes in the individual root-to-leaf paths as Harmonygram icons, underlain by the number of times each chord appears in the tree branch under consideration. Additional symbols, which are easy to recognize, are used for rests, as well as for song and phrase starts and endings. This concept is illustrated in Figure 7.

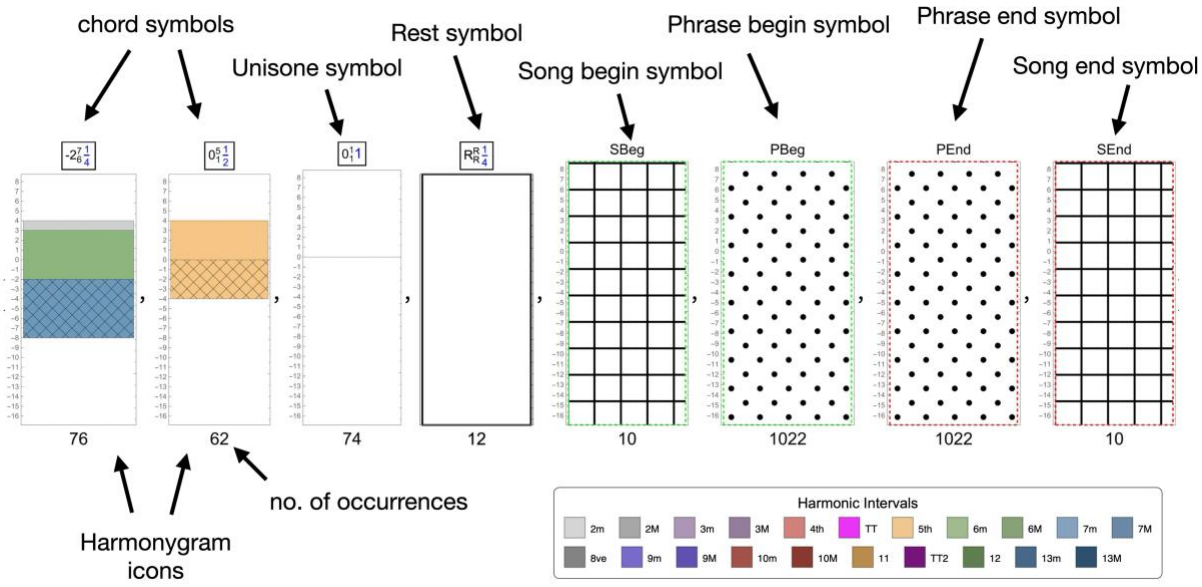

Figure 7. Representation of the nodes of the root-to-leaf-paths as Harmonygram icons with additional information. The icon label explains each icon verbally while the number below the icon indicates the number of times a particular chord or symbol appears in the root-to-leaf-path considered (here randomly assigned for illustration only).

As an implementation of this concept, Figure 8 shows the "root-to-leaf-path representation" of the 30 most frequently used deterministic production rules in the Erkomaishvili corpus using the Harmonygram icon representation.

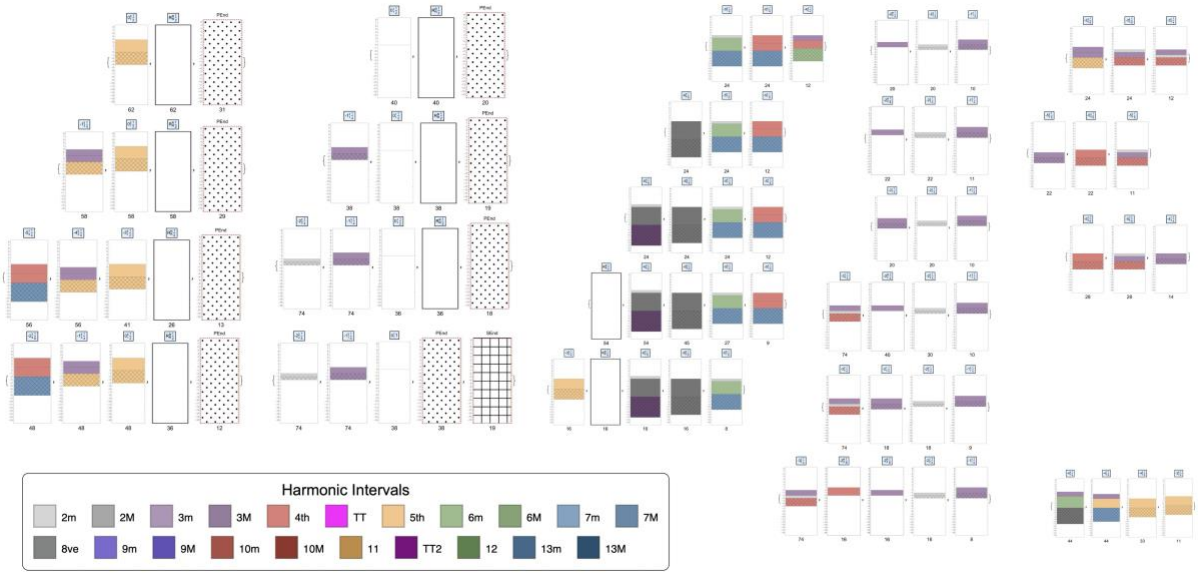

Figure 8. Representation of the nodes of the root-to-leaf-paths of the 30 most often used deterministic production rules in the Erkomaishvili corpus in Harmonygram icon representation.

Figure 8 demonstrates that the complete information in Figure 6 can now be visually perceived instantly, effectively making it a 'fast process' in the Kahneman sense (Kahneman, 2013). It also becomes immediately apparent that relationships between the individual production rules exist, and they can be grouped into six different categories with similar chord progression characteristics. We refrain from further interpretation at this point and move on to the discussion of cadences.

What are typical cadences?

Figure 9 shows the most frequently used cadences in the Erkomaishvili corpus.

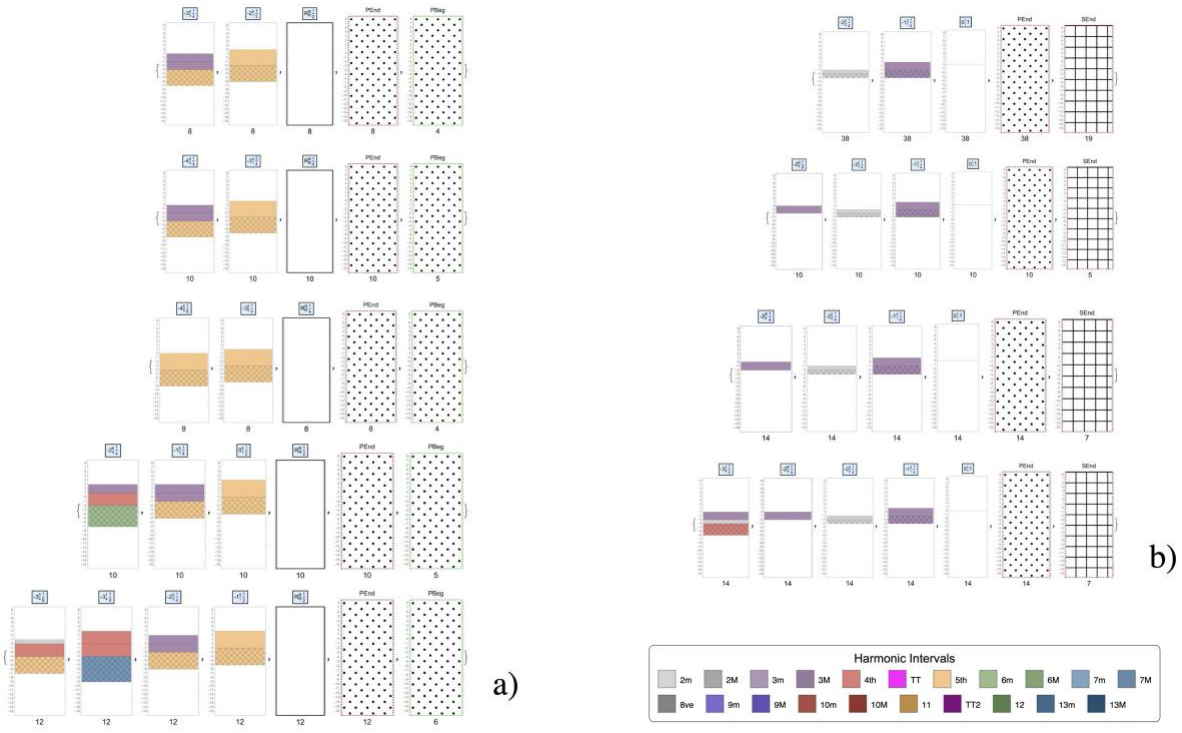

Figure 9. Most often used phrase cadences (a) and song cadences (b) in the Erkomaishvili corpus in Harmonygram icon representation.

It is evident that phrases most often end on a fifth, followed by a rest. This contrasts with song cadences, which typically end on unison.

Using the production rules as a generative model

Finally, one of the most natural applications of the production rules derived from the Kohonen model is to use them as generative models to create new songs similar to (Sheikholharam and Teshnehlab, 2008). Figure 10 shows six such examples as piano roll displays.

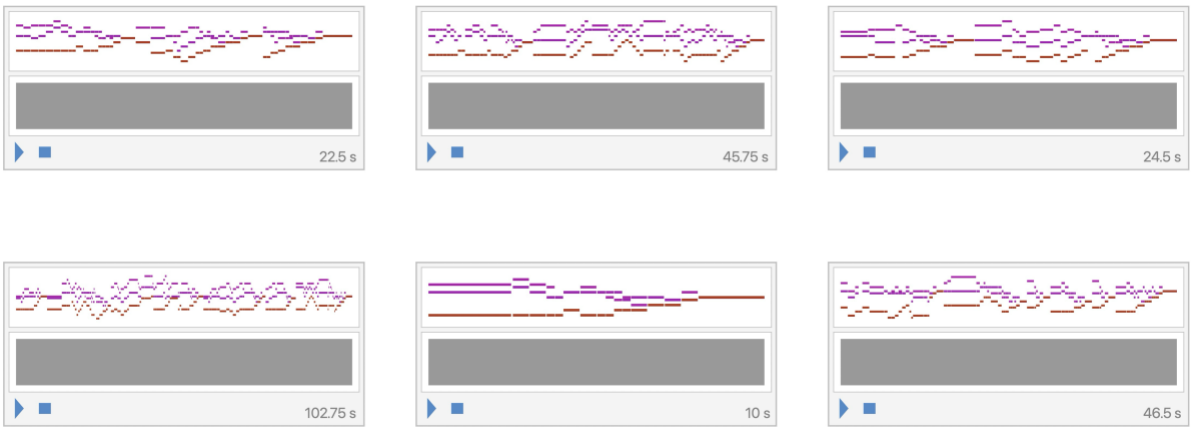

Figure 10. Piano-roll display of 6 synthetic songs, generated from prefix tree of the production rule set of the Erkomaishvili corpus.

In these examples, one can observe the complexity of the voicings and the development of the coda, often characterized by a typical stepwise upward movement in the bass voice, which frequently concludes in unison with the middle and top voices. However, sometimes the model seems to get stuck temporarily in very repetitive patterns, until it finds a way out of it, as seen in the lower-right example.

Can we benefit from the recent developments in AI research?

Towards the end of 2022, new developments in computer science, particularly the public availability of OpenAI's ChatGPT model, generated significant excitement within the scientific community and beyond. By now, discussions about the possible implications of these developments have also reached the field of ethnomusicology (Morales et al., 2024). In the following, we will discuss what we currently[4] believe could be concrete consequences for our work.

ChatGPT is a specialized artificial neural network (ANN) designed to model language. For our purposes, it suffices to understand language models (LMs) simply as probabilistic models trained to perform one task: predict the next word in a sentence based on the preceding words. In this sense, the prefix tree representation of the Kohonen grammar used earlier to generate synthetic songs, as shown in Figure 10, is also a language model, albeit a very small one. In contrast, at the core of ChatGPT is a Large Language Model

[4] Since this field is developing at an astonishing rate and new tools appear in rapid succession, please note that what is considered as the best approach today may be obsolete tomorrow.

(LLM), which is trained on vast amounts of data and possesses some initially surprising properties. These properties arise because, during their training process, LLMs do not only learn the syntax rules of a language, which they use to complete sentences, but also acquire 'semantic concepts' or 'meanings'. This phenomenon is related to the concept of 'embeddings', which will be explained below.

Within a neural network trained on text, each word or token is represented by a long list of numbers, known as a feature vector. These vectors can be imagined as 'points' in a high-dimensional 'feature space', also referred to as the 'embedding space'. The numbers initially assigned to a word or token in this space are unique, but their actual values don't have any intrinsic meaning. However, during training, as the network learns to predict the next words, the numerical values of these feature vectors change. Once training is complete, words that are somehow 'similar in meaning' also end up being close to each other in this embedding or feature space. As Stephen Wolfram beautifully illustrates in his blog[5] *What is ChatGPT Doing—and Why Does It Work?*, which is also available as a book (Wolfram, 2023), an embedding can be thought of as a way to represent the 'essence' of something by lists of numbers, with the property that 'similar things' are represented by lists with similar numbers.

To visualize these high-dimensional feature vectors, they must be projected into two dimensions. In Stephen Wolfram's blog (mentioned above), one can observe how words corresponding to different parts of speech are laid out in an embedding. For example, nouns, verbs, adjectives, and adverbs are well separated.

Driven more by curiosity than by a justified conviction that it would work, we decided to explore whether we might also detect some 'linguistic relations' through the visual inspection of the embedding space of an artificial neural network trained on our data. To address the fact that the size of our corpus is far too small to train any artificial neural network directly, we used a technique called 'data augmentation'. For the training data, we used the Erkomaishvili dataset and restricted ourselves to the deterministic rule-set of the Kohonen grammar model as discussed earlier. We employed a recurrent neural network (LSTM network) trained on 80,000 synthetic songs generated from this rule set,

[5] https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/

similar to the examples shown in Figure 10.

Figure 11a shows the chord embedding map for all the chords in the Erkomaishvili corpus before the network was trained. Consequently, the spatial distribution is random, reflecting how these vectors were initially initialized. After the network was trained — meaning it had learned to complete chord sequences— the spatial structure was no longer random (Figure 11b). However, it was still difficult to assign any specific interpretation to it.

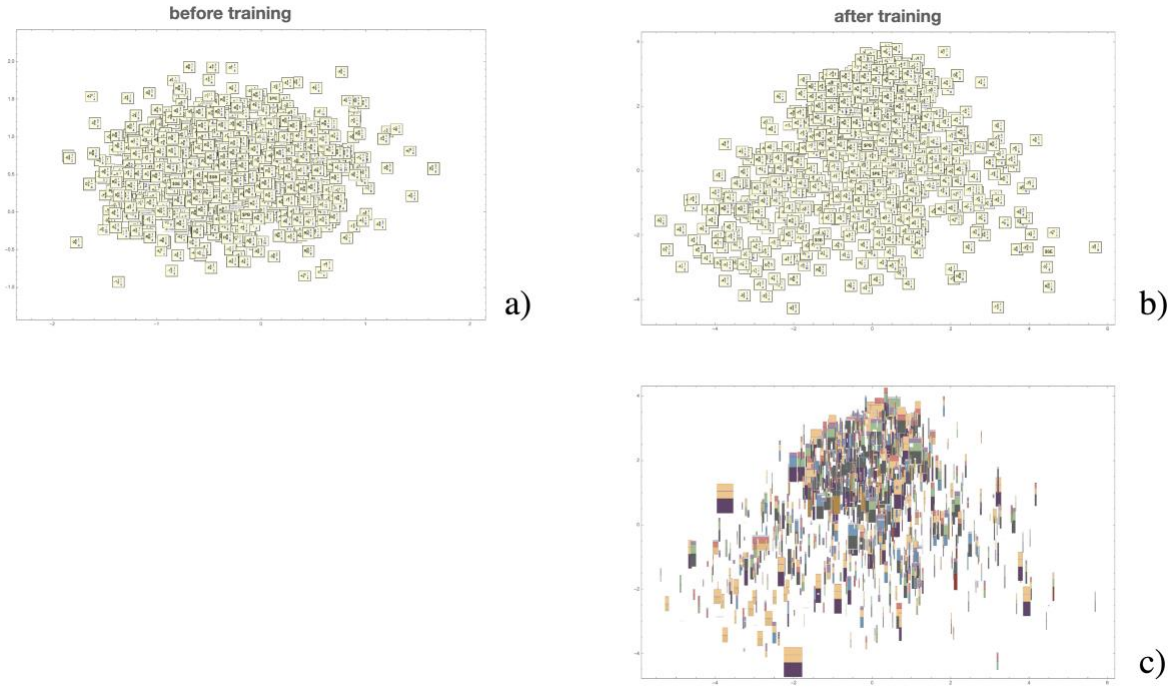

Figure 11. Chord embeddings for all chords in the Erkomaishvili dataset, based on the deterministic rules occuring more than once in the Kohonen grammar model.

When all the chords are shown as Harmonygram icons (Figures 11c and 12), the embedding map becomes slightly more informative. Figure 12 presents this map in higher resolution.

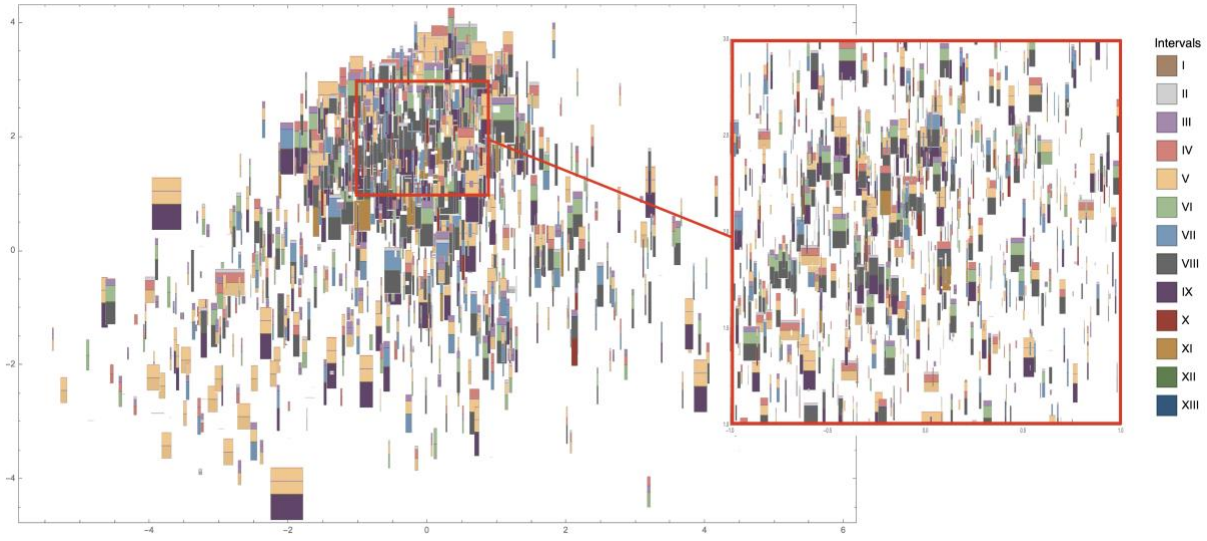

Figure 12. Same as Fig. 11 c), but in higher resolution for better visibility.

In Figure 12, one can observe that the chords in the upper central part of the embedding map generally have longer durations and display greater diversity in type compared to the majority of the chords in the lower left. Given the still relatively small dataset size (in comparison to the massive data volumes used for training Large Language Models or LLMs), we should avoid overinterpreting this map. However, it is fair to say that the map is clearly structured and less random. For instance, in the lower left, we see many single interval chords forming a fifth (represented by sand-colored bars with a horizontal line in the middle). In the Erkomaishvili corpus, these chords frequently appear as segment cadences (cf. Figure 9a). This could explain their distinct positions in the embedding map due to their different functional roles in the song structure.

We want to emphasize that the remarks above should be seen more as speculations than as conclusions. What we feel we can take away from these plots at this point in time is that there is some structure in the embedding map that might be meaningful, but we are not yet at a stage where we can interpret it in a sound way. This may be due to the limited size of the dataset, but it could also suggest that the assumption of 'linguistic' relationships between individual chords, which could be inferred through visual inspection of the embedding space, might not be applicable —at least not for the chords we are interested in.

But what about differences in the syntax of individual songs? Could we detect these in the

embedding map of the songs, as opposed to the embedding maps for chords? Our complete corpus consists of a total of 452 songs, which can be categorized into different classes. Some of these classes are regional, while others are associated with specific liturgical schools, such as Shemokmedi and Gelati, two monasteries in western Georgia. From a recent classification study, we know that the chord progression sequences in the songs differ across these classes (Scherbaum et al., 2024).

As a final experiment, we trained a Generative Transformer Network—essentially the same type of model that powers ChatGPT—using the complete corpus. To address the challenge of our corpus's small size, we again employed 'data augmentation,' following a similar approach as before. Since we know all the production rules that define our corpus according to the Kohonen model, we can generate an unlimited number of additional synthetic songs and assume that these correspond to songs that simply have not yet been produced. These synthetic songs were then combined with the real ones.

We skip the technical details, which are irrelevant to the following discussion, and jump straight to the resulting embedding map, shown in Figure 13.

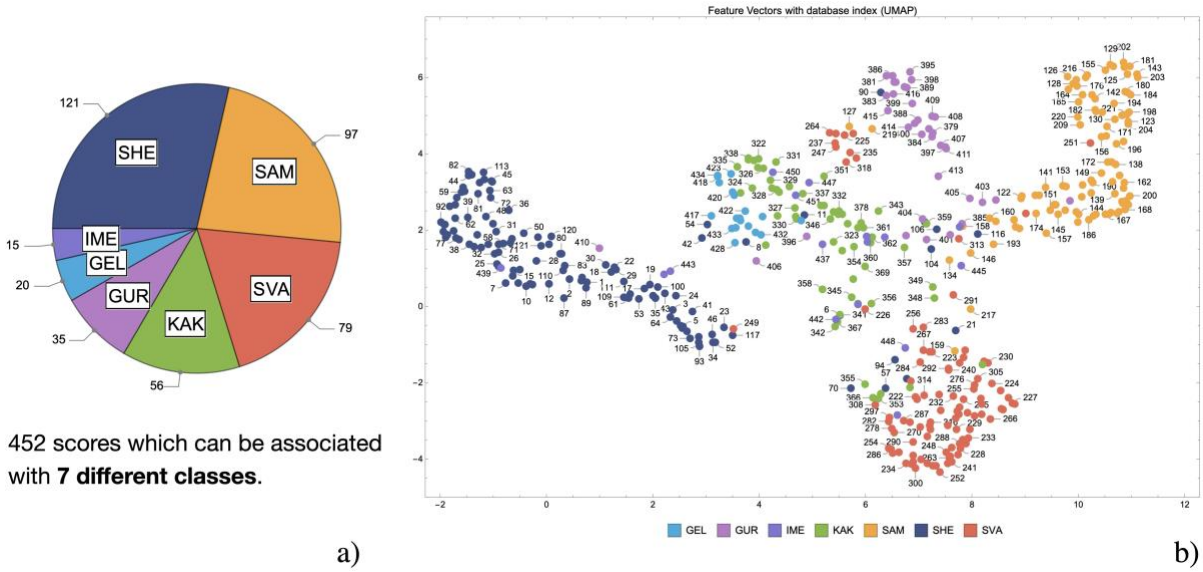

Figure 13. Feature map (embedding) for the songs in the complete corpus. Fig. 13 a) shows the partitioning into different classes. Fig. 13b) displays the individual feature vectors projected into 2 dimensions. The acronyms IME, GEL,GUR, KAK, SVA, SAM, and SHE stand forImereti, Gelati, Guria, Kakheti, Svaneti, Smaegrelo, and Shemokmedi, respectively.

Each point in Figure 13b represents a song, while the different colors correspond to different classes or subsets within our corpus. You can see that the different classes are

well-separated in this map[6]. This suggests that in general the temporal chord progression structures between these classes are systematically different. In other words, the network effectively captures the differences between the individual subsets of the corpus. It is worth noting that, in contrast to the n-gram based classification (Scherbaum et al., 2024), this differentiation was achieved in a completely unsupervised manner.

Discussion and conclusions

The interdisciplinary project presented here has come a long way, overcoming many challenges along the way. The ultimate goal, the quantitative description of the chordal syntax of traditional Georgian vocal music, still lies ahead of us. However, we believe that with the results presented here, we have reached an important intermediate milestone. The self-learning Kohonen model represents a significant improvement over the classical Markov model, as it allows for the influence of different context lengths for each chord in a chord sequence. It provides a complete description of the production rules governing chord sequences, assuming that only the immediate context of a chord is relevant for its generation. With the representation of the Kohonen model in the form of prefix trees, we now have an efficient generative model that allows for the generation of an unlimited number of synthetic songs, all adhering to the rules of the dataset used to train the model.

Technically, the limitation to the immediate context can now be overcome using artificial neural networks based on transformers, as employed in models like ChatGPT. However, the amount of data necessary for their training far exceeds what is typically available in ethnomusicological research. This is certainly true for traditional Georgian vocal music, and it is unlikely that this limitation will ever be overcome. Therefore, it is probably fair to say that the current model has methodologically achieved what —at least with today's methods— can be extracted from musical notation, which also answers the question posed by the title.

Our project has provided us, in addition to the generation of the Kohonen model, with several important insights. As an interdisciplinary team without a common theoretical background, we have come to appreciate the great importance of visual analyses, such as

[6] Individual songs represented by points whose color does not match that of adjacent points appear to deviate from this pattern. The reason for this will have to be investigated in detail, but this will have to be done outside the scope of this work.

the representation of songs as Harmonygrams and chords as Harmonygram icons, as these are intuitively understandable.

From our perspective, two important tasks remain to be addressed next. First, the Kohonen model represents every part of a song as equally important, meaning it does not differentiate between ornamental and grammatically essential aspects. As a result, the model is somewhat 'overloaded' and difficult to interpret. We are still in the early stages of addressing this problem (Arom and Scherbaum, 2024). Secondly, all models have so far been culturally unvalidated. This means that their results have not yet been tested by an audience familiar with Georgian music, and their ethnomusicological relevance remains to be evaluated.

Acknowledgments

Part of this work was supported by the German Research Foundation (DFG MU 2686/13-1,SCHE 280/20-1).

REFERENCES

Antonov, Anton. (2018). Tries With Frequencies Mathematica package. Retrieved from https://github.com/antononcube/MathematicaForPrediction

Antonov, Anton. (2023). Using Prefix trees for Markov chain text generation. Retrieved from https://community.wolfram.com/groups/-/m/t/2819012

Arom, Simha; Scherbaum, Frank. (2024). *Towards a Theory of the Chord Syntax of Georgian Polyphony: New Tools, New Perspectives and New Results.* Paper presented at the 12th International Symposium on Traditional Polyphony: Tbilisi, Georgia, 25-28 September, 2024.

Arom, Simha; Scherbaum, Frank and Darras, Florent. (2018). "Structural Analysis and Modeling of Georgian and Medieval Polyphonies." *Proceedings of the 9th International Symposium on Traditional Polyphony.* [9. International Symposium on Traditional Polyphony, Oct 30 - Nov 3, 2018, Tbilisi, Georgia] Eds. Rusudan Tsurtsumia and Joseph Jordania: pp. 203–319. Tbilisi: International Research Center for Traditional Polyphony

of Tbilisi State Conservatoire.

Arom, Simha; Vallejo, Polo. (2008). "Towards a theory of the chord syntax of Georgian Polyphony" *Proceeding of the 4th International Symposium on Traditional Polyphony.* [The Fourth International Symposium on Traditional Polyphony]. Eds. Rusudan Tsurtsumia and Joseph Jordania: pp. 321–335. Tbilisi: International Research Center for Traditional Polyphony of Tbilisi State Conservatoire.

Arom, Simha; Vallejo, Polo. (2010). "Outline of a syntax of chords in some songs from Samegrelo" *Proceeding of the 5th International Symposium on Traditional Polyphony.* [The Fifth International Symposium on Traditional Polyphony] Eds. Rusudan Tsurtsumia and Joseph Jordania: pp. 266–277. Tbilisi: International Research Center for Traditional Polyphony of Tbilisi State Conservatoire.

Kahneman, Daniel. (2013). *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.

Killick, Andrew. (2021). *Global Notation as a Tool for Cross-Cultural and Comparative Music Analysis*. Retrieved from https://eprints.whiterose.ac.uk/id/eprint/151037/3/Killick_AAWM_Vol_8_2.pdf

Kohonen, Teuvo. (1989). "A self learning musical grammar, or "Associative memory of the second kind" *International Joint Conference on Neural Networks.* 1: pp. 1-5, Washington: International Neural Network Society.

Morales, Glinore S.; Perez, Mary Leigh Ann C. and Tabuena, Almighty C. (2024). "Artificial Intelligence and the Integration of the Industrial Revolution 6.0 in Ethnomusicology: Demands, Interventions and Implications" *Musicologist*. 8(1): 75-107. Retrieved from https://dergipark.org.tr/en/pub/musicologist/issue/85473/1286472

Rosenzweig, Sebastian; Scherbaum, Frank; Shugliashvili, David; Arifi-Müller, Vlora and Müller, Meinard. (2020). "Erkomaishvili Dataset: A Curated Corpus of Traditional Georgian Vocal Music for Computational Musicology" *Transactions of the International Society for Music Information Retrieval*. 3(1): 31-41. Retrieved from https://doi.org/10.5334/tismir.44

Scherbaum, Frank. (2024). *Going Beyond Western Scores: An Alternative Notation System*

for Traditional Georgian Vocal Music. Paper presented at the 12th International Symposium on Traditional Polyphony. Tbilisi, Georgia. 25-28 September, 2024.

Scherbaum, Frank; Arom, Simha; Caron Darras, Florent; Lolashvili, Ana and Kane, Frank. (2024). "On the Classification of Traditional Georgian Vocal Music by Computer-Assisted Score Analysis" *Musicologist*. 8(1): 28-54. Retrieved from https://dergipark.org.tr/en/pub/musicologist/issue/85473/1246886

Scherbaum, Frank; Arom, Simha and Kane, Frank. (2015). "On the feasibility of Markov Model based analysis of Georgian vocal polyphonic music" *Proceedings of the 5th International Workshop on Folk Music Analysis, [The 5th International Workshop on Folk Music Analysis].* (pp. 94-98). *Paris: University Pierre and Marie Curie*

Scherbaum, Frank; Arom, Simha and Kane, Frank. (2016a). "A graph-theoretical approach to the harmonic analysis of Georgian vocal polyphonic music" *Proceedings of the 6th International Workshop Folk Music Analysis,* [The 6th International Workshop Folk Music Analysis] Beauguitte, Pierre; Duggan, Bryan; Kelleher, John D. (Eds.), (pp. 59-60). Dublin: Technological University.

Scherbaum, Frank; Arom, Simha and Kane, Frank. (2016b). "Graphical comparative analysis of the harmonic structure of the Akhobadze corpus of Svan songs" *Proceeding of the 8th International Symposium on Traditional Polyphony*. [Paper presented at the 8th International Symposium on Traditional Polyphony on September 26-30, 2016, Tblisi, Georgia]. Tsurtsumia, Rusudan and Jordania, Joseph (Eds.), (pp. 170-189). Retrieved from https://drive.google.com/file/d/1tQ_qDqY5xLWpOF8S3Ns54_ChCb_Wf72D/view

Scherbaum, Frank; Mzhavanadze, Nana. (2024). "Harmonygrams: A Graphical Notation System for Three-Voiced Music Facilitating the Perception of Harmonies" [Paper presented at the Eighth International Conference on Analytical Approaches to World Musics (AAWM 2024). Bologna, Italy, June 10-14, 2024]. Retrieved from https://youtu.be/8pWI9z_SSUM

Scherbaum, Frank; Mzhavanadze, Nana; Arom, Simha; Rosenzweig, Sebastian and Müller, Meinard. (2020). Tonal Organization of the Erkomaishvili Dataset: Pitches, Scales, Melodies and Harmonies. Potsdam: Universitätsverlag Potsdam.

Scherbaum, Frank; Mzhavanadze, Nana; Arom, Simha; Rosenzweig, Sebastian and Müller, Meinard. (2021). *Analysis of Tonal Organization and Intonation Practice in the Tbilisi State Conservatory Recordings of Artem Erkomaishvili of 1966.* [Paper presented at the Sixth Analytical Approaches to World Music Conference, June 9-12, 2021, Paris, France]. Special Session in Honor of Simha Arom. Retrieved from https://www.youtube.com/watch?v=tfjy_q71WUQ

Scherbaum, Frank; Mzhavanadze, Nana; Arom, Simha; Rosenzweig, Sebastian and Müller, Meinard. (2023). "Tonal Organization of the Erkomaishvili Dataset: Pitches, Scales, Melodies and Harmonies". *Anzor Erkomaishvili and Contemporary Trends in the Study of Traditional and Sacred Georgian Music*, Eds. Jordania, Joseph and Tsurtsumia, Rusudan: pp. 53-88. Cambridge: Cambridge Scholars Publishing.

Scherbaum, Frank; Mzhavanadze, Nana; Rosenzweig, Sebastian and Müller, Meinard. (2022). "Tuning Systems of Traditional Georgian Singing Determined From a New Corpus of Field Recordings" *Musicologist*. 6(2): 142-168. Retrieved from https://dergipark.org.tr/en/pub/musicologist/issue/74133/1068947 .

Sheikholharam, Peyman; Teshnehlab, Mohammad. (2008). "Music Composition Using Combination of Genetic Algorithms and Kohonen Grammar | IEEE Conference Publication | IEEE Xplore" *2008 International Symposium on Computational Intelligence and Design*. Retrieved from https://ieeexplore.ieee.org/document/4725603

Shugliashvili, Davit. (2014). Georgian Church Hymns, Shemokmedi School. Tbilisi: Georgian Chanting Foundation & Tbilisi State Conservatory.

Tsereteli, Zaal; Veshapidze, Levan. (2014). "On the Georgian traditional scale" [The Seventh International Symposium on Traditional Polyphony] Eds. Rusudan Tsurtsumia and Joseph Jordania: pp. 288–295. Tbilisi: International Research Center for Traditional Polyphony of Tbilisi State Conservatoire

Tsereteli, Zaal; Veshapidze, Levan. (2015). "Video of the presentation "The empirical research of a Georgian sound scale" 2015 IAML/IMS Congress. New York City, USA.

Wolfram, Stephen. (2023). *What Is ChatGPT Doing and Why Does It Work?* Illinois: Wolfram Media.