Analysis of the structure similarity of musical scores

Andrei-Daniel Ghițeanu
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
ghiteanuandrei@gmail.com

Ștefan Trăușan-Matu
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
and
Research Institute for Artificial Intelligence
and
Academy of Romanian Scientists
stefan.trausan@upb.ro

ABSTRACT
The field of music analysis has expanded significantly in response to recent developments in artificial intelligence and machine learning. The research presented herein is focused on the study of resemblance between musical scores, therefore exploring different techniques used in calculating similarity ratings. The processing starts by representing the musical data extracted from song files in a manner that facilitates analysis and presents advantages in depicting the relationships between the elements present in the compositional structure of the musical pieces, leading to the use of a knowledge graph for this purpose. Three proposed methods will be inspected for calculating the ratings of comparison between different melodies and music measures, based on generating embeddings, music criteria and integration. The aim is to verify the reliability for integration of those approaches while developing more complex applications.

Author Keywords
Musical score analysis, ontology, music similarity, artificial intelligence.

ACM Classification Keywords
H.5.1. Multimedia Information Systems: Audio input/output; Applied computing → Sound and music computing; Computing methodologies → Artificial intelligence; Machine learning.

DOI: 10.37789/ icusi.2024.18

INTRODUCTION

Despite the fact that music existed long before music theory, it is nonetheless crucial for comprehending the structure of music as it clarifies the connections between notes, chords, and songs. These activities improve the compositional and analytical abilities of musicians and composers, such as musical hearing or improvisation.

Researchers and practitioners have long been interested in musical creation, and over time, several approaches and methodologies have been created to study and comprehend various aspects of music. The use of these techniques was however limited by the difficulty of handling and extracting features from the symbolic form of data, such as musical score [1]. The study focuses on music similarity, with the goal of investigating certain approaches for accurately comparing and contrasting musical works. One purpose is to construct algorithms capable of finding and measuring the similarities and differences between musical scores, considering various musical components such as melody, harmony, rhythm, and structure. There are various reasons why studying music similarity is essential. It can improve instructional tools by helping students to learn from similar instances and comprehend how diverse compositions can be linked by common musical concepts.

Moreover, it facilitates the detection of musical plagiarism by identifying highly comparable pieces, thus ensuring the integrity and originality of musical creations. Additionally, music similarity analysis can improve music recommendation systems by supporting listeners in discovering new music that matches their preferences based on structural and stylistic similarities.

Among the key concepts of this field, the identified features are extraction, representation, similarity, and transcription of music, which can be used for improving automated music analysis, new music generation and music understanding.

- The data extracted from the music files must be suitable for score analysis. Multiple elements can be recognized in the construction of a song, including four compositional principles that could be taken into consideration, represented by melody, rhythm, sound, and harmony, and used as starting points for the extraction of features.
- Music representation refers to how music is reproduced in a digital format such as MIDI (Musical Instrument Digital Interface) files, audio recordings, scores, or symbolic representations such as MusicXML, which will be described in the upcoming sections. Choosing the right format can significantly affect the performance of machine learning algorithms that use music data from extracted features.
- Music similarity is the process of using various features to measure how similar two or more pieces of music are.
- Music transcription involves converting audio recordings of music into a symbolic representation, such as musical scores or MIDI files.

STATE OF THE ART

Musical Instrument Digital Interface (MIDI) is a form of storing audio files in digital form. This interface can be used for various purposes, with numerous applications such as connecting compatible devices together. For example, a connection can be established between two synthesizers [4], or to link a synthesizer to certain software applications. MIDI also allows for the storage of melodies in digital format, a representation that provides the possibility of training machine learning models due to the information related to compositions (such as notes or timings) extracted from these types of files.

MusicXML is another format for representing music, based on Extensible Markup Language (XML). It was based on two academic formats, MuseData and the Humdrum **kern format. Its development took place alongside the development of MusicXML software, and the first software prototype performed two-way conversion with MuseData, reading from NIFF (National Interchange File Format) files and writing to standard MIDI files [5]. It can be used to play music in a highly detailed manner and can represent a wide range of musical notation styles.

MusicXML is also a format used for analysis and the utilization of machine learning algorithms. Artificial intelligence techniques, along with machine learning algorithms, are combined with advanced mathematics and programming techniques, such as constraint programming or genetic programming.

With the introduction of the music21 package, used in Python, both musicians with little programming experience and programmers without a deep understanding of music theory can make use of very useful tools that have integrated musical knowledge [2]. It provides a modular approach that combines object-oriented programming with a simple interface. With such capabilities, representations, analyses, and manipulations of data can be carried out in asymbolic form, in this case represented by musical scores. This toolkit can be used to establish a connection between the advanced study of musical compositions and the web environment. A software architecture based on Service-Oriented Architecture (SOA) is proposed, making it possible to integrate more complex methods into applications [3].

Constraint programming

Compositional rules describe music in such a way that it can be divided into independent or interchangeable pieces. These rules do not exactly describe how a melody should be composed; instead, they describe important features that should be found in the final piece. A programming paradigm that can be used to address problems related to certain rules is constraint programming [6]. This approach introduces techniques used to solve Constraint Satisfaction Problems.
These types of problems consist of a set of variables and mathematical relationships defined between them. There are known systems such as PWConstraints (originally developed as a library on top of PatchWork [22] for computer-assisted composition), Situation (a system for solving problems related to harmonies, originally developed as a library for PatchWork), MusES [23] and BackTalk [24] (an expressive and extensible system applied for automatic harmonization), OMClouds [25], and Strasheela [26].

Genetic programming

For problems involving the identification and optimization of parameters and structures to find solutions, Genetic Programming (GP) can be a suitable approach. Different equations can be written in the form of tree-like structures, where the leaves are treated as input parameters or constants, and the rest of the nodes represent mathematical operations. Various combinations of data can be tried to obtain a favorable result, and through a selection process, the optimal solution is returned.

GP is based on the propagation of generations through a selection process. The first generation is populated by randomly chosen individuals (structures), each evaluated along with its performance [7]. This can be related to the research performed by Andres and Inden [8]. The program creates a homorhythmic series (same rhythmic values) of chords, where a new chord starts at the same time as other notes in the score and contains all the notes from that moment. When a dissonant chord is found, it checks if it becomes consonant by removing the note that created the dissonance. The algorithm progresses in an order dependent on the duration of the notes. The input data for the machine learning part, in addition to notes and rhythmic values, also contains a knowledge base (such as melodic intervals and accent weights).

Dissonances are classified using the DBSCAN algorithm. For learning the rules of each dissonance category, a learning example is added, consisting of a set of three notes with a dissonant note in the middle. STGP (Strongly Typed Genetic Programming) is used, implemented in the DEAP library for Python. The nodes in the tree representing a learned rule can be logical operators (∨ - or, ∧ - and, ¬ - not, → - implies, ↔ - equivalent), arithmetic or relational operators (+, -, *, /, <, >, =), or conditional expressions (if_else). The leaves are considered constants, true or false values, or integers between 0 and 13, or input variables such as the duration of the dissonant note and the two notes before and after it, as well as the intervals between the dissonant note and the preceding and succeeding notes, along with the accentuation of the dissonance.

DATA REPRESENTATION

Depending on how complex and interconnected the data is, certain approaches may be more appropriate for future uses or operations. One of the factors that can make the operation and handling of data more efficient is represented by the organization of a database. Frequently, this step can be challenging, and depending on the data that must be preserved, choosing the proper type of database can significantly affect how that data is used.

A musical composition can include multiple parts, each of which can have a continuous stream of notes or chords. All these components may have different attributes, resulting in the observation of a particular hierarchy in a score's structure. Furthermore, each of these components can be represented individually and then be related to one another through explicit relationships. Every part, in addition to specific attributes, has in its composition a stream of notes at different intervals, also considering the pitches of each note. Chords represent two or more notes that will be played at the same time during a song. These can be extracted either by finding them in the individual parts of a piece or by overlapping those parts.

Ontologies and knowledge graphs

An ontology describes a formal representation of concepts, entities, and relationships that captures knowledge in a specific domain [13]. A deeper and more complex representation of knowledge is possible by creating semantic nets, which connect and associate concepts.

They are frequently constructed in a form that enables the storage, retrieval, and efficient manipulation of knowledge and allows for the expression of advanced knowledge systems, including classifications and hierarchies.

A lot of interest was generated as they are useful in activities such as data analysis and machine learning. They provide a method for representing interactions between items with the help of a knowledge graph, which is a structure used to add data in the form of individuals and establish meaningful connections while facilitating information extraction and analysis.

While an ontology is used to define classes and properties, the purpose of the graph is to present their instances. The capabilities of this graph include efficient querying, reasoning, and inference, which leads to advanced analysis. “The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web." which is based on documents that use RDF models [9], used to represent data in the form of triplets (subject, predicate, object). RDF vocabularies (RDF Schema - RDFS) and Web Ontology Language (OWL) are used to describe entities and the relationships between them. This process was performed using the RDFLib package for Python.

Music Ontologies

Consideration of an ontology's expansion is an essential aspect of its development. However, to establish alignment, the current ones must be properly observed to identify classes and relationships that can be used.

The Music Ontology [10] is a model of an existing ontology for representing music-related data, which includes editorial, cultural, and acoustic information. Three other ontologies on which it is built are Timeline Ontology, Event Ontology and Functional Requirements for Bibliographic Records ontology (FRBR) [14]. Nevertheless, the capability of The Music Ontology is highlighted by rendering data related to certain general information about compositions.

The HaMSE Ontology (Harmonic, Melodic, Structural and Emotional features ontology) [11] and The Music Note Ontology [12] contain classes and properties defined in a certain way that they can be used for data representation. The structure of these ontologies had an impact on how the concepts and attributes were defined in the one created in this stage of the research.

The ontology used for this research was designed in such a way that it satisfies the necessary conditions for its subsequent use without extending any existing models. The musical elements, connections and attributes can be seen in Figure 1.

Feature Extraction

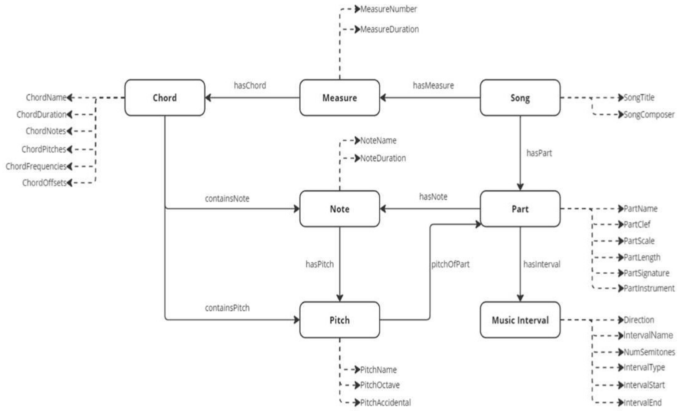

Figure 1. Ontology schema

The extractor was implemented using the available methods from the music21 package. The extractor can get each string of elements within each structure and then access various attributes of those components. The source of the music files is ChoralWiki, which represents the Choral Public Domain Library.

Additionally, some hierarchical construction observed in musical compositions is important to keep data extraction simple. Because of this composition of the score, it is possible to navigate and access the elements according to their location in the hierarchy, gaining access to the important characteristics of the elements during the operation. This was useful for obtaining features such as parts, notes, pitches, and intervals. The chords were obtained by overlapping parts of the song, included on different measures, and extracted according to them. For each of the mxl files, a new JSON file was generated, which contains a dictionary created after the feature extraction process. For the serialization of this file, a custom encoder was used to save the attributes of the music21 objects. These will be accessed afterwards for reading and creating URIs for adding instances to the graph.

Visual representations of the relationships between individuals can be observed by using GraphDB, which is a database management system developed by Ontotext. It is also used in a variety of applications, such as data integration, semantic search, and knowledge management. An example of such visualization can be seen in Figure 2, where an expansion was made from a Song node towards the elements of a chord, through the song’s measures

MUSIC SIMILARITY

Finding score similarity based on compositional structure is one branch of music score analysis that is worth investigating. As the field progresses, collaborative efforts between data scientists, computer scientists, and musicologists are essential for its advancement. Structural similarity can have multiple applications, offering the possibility to develop more advanced software to perform certain operations, such as music recommendation based on compositional elements, detection of plagiarism for new composed scores, or to provide aid in learning music theory by offering more examples that are similar to the one that is studied by the user.

Machine learning using graphs, also known as graph-based machine learning, has become a strong paradigm with different applications in several disciplines.

The ability of graph-based machine learning to effectively simulate complex relationships and dependencies that may be difficult for conventional models to understand represents one of its main advantages. Graph algorithms facilitate the recognition of patterns and clusters among interconnected data, offering a comprehensive understanding of the fundamental structure. The capacity to extract and utilize the large amount of information included in graph structures can also improve the accuracy of a trained model. This type of structure is suitable for musical scores because it allows interactions between elements to be represented through distinct individuals, defining a hierarchy, and avoiding redundancy while precisely capturing the intricate structure of a piece. Therefore, methods that can be used for training with learning based on the defined connections inside the graph are preferred.

When comparing two musical scores, an in-depth examination considering the different components, methods, and structures used in the creation is required. The overall comparison is influenced by various features of music, and the criteria used can change based on the analysis's particular objectives and circumstances. Melody and rhythm are two fundamental components of a musical piece.

Conducted research towards the detection of music plagiarism examined different methods that can be used to determine music similarity, such as spectrogram analysis [17], similar melody searching based on alignment and shifting [18], and computational intelligence modules that include unsupervised machine learning and a fuzzy deep analyzer [19].

METHODS

Three approaches can be evaluated to determine a similarity rating between melodies as well as between parts of musical pieces (based on comparing musical measures). One is based on training a model on the knowledge graph to generate embeddings, and the others are based on criteria, respectively, on integration, to determine the resemblance between measures.

Embeddings

For this method, Node2Vec was chosen to be trained on the collected data, which is "an algorithmic framework for learning continuous feature representations for nodes in networks" used for multi-label classification and link prediction [20]. It considers each node as an individual word, and random walks are considered to be sentences. The resulting embeddings should reflect not only the immediate connections of each node, but also their broader context within the network, leading to a more comprehensive understanding of the graph's structure.

There are two types of similarities that are considered when embedding the nodes closely together. The first one is homophily, which refers to the connections between nodes in a network, and the second one represents structural equivalence, considering the presence of the node and denoting how similar its connectivity pattern is to others inside the network. For homophily, the connectivity of nodes matters, while for structural equivalence, it does not, observing the structural role at any distance inside the graph. The model provides a representation encapsulating the characteristics of both aspects.

One disadvantage of the Node2Vec algorithm is that it does not count node characteristics, such as duration for note nodes or part details. Still, because of the structure of the knowledge graph and how the nodes are connected, this framework should be able to consider as many items as possible. When adding individuals, such particularities are added as literals, and when parsing the ontology file, the input graph created for Node2Vec will include them as nodes. Consequently, those will be part of the network and should contribute to the overall analysis conditions.

Criteria-based approach

Based on measure properties and elements, a formula can be defined to calculate a similarity rating (Sm), represented by a sum of the products between different similarity factors of two measures and a corresponding weight.

$$S_m = \sum_i W_i * s_i$$

There are four identified factors on which the resemblance is based on: the number of identical chords (s_c), the length of the measure (s_{ml}), the rhythm (s_r), given by the duration of the chords, and the pitch distribution (s_{pd}). The weights for each coefficient will be determined as $W_c = 0.4$, $W_{sml} = 0.1$, $W_r = 0.3$, and $W_{pd} = 0.2$, depending the relative importance of each factor, depending on the possible impact that it has on the similarity rating (e.g., two measures that contain the same chords, almost in the same order, but have a different duration, can be perceived as more comparable).

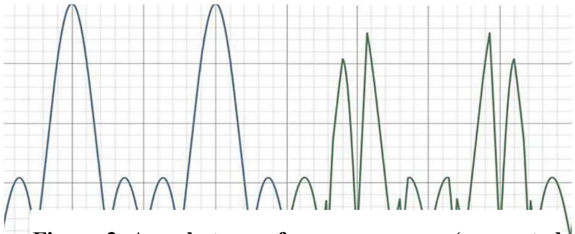

Figure 3. Area between frequency waves (generated using: https://www.desmos.com/calculator)

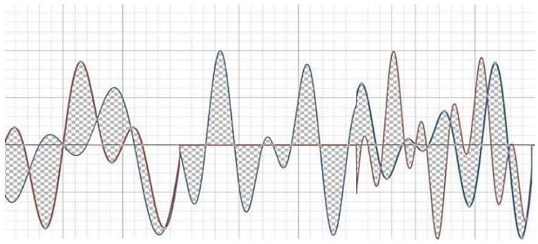

Figure 4. Effect of absolute value applications on the functions (generated using: https://www.desmos.com/calculator)

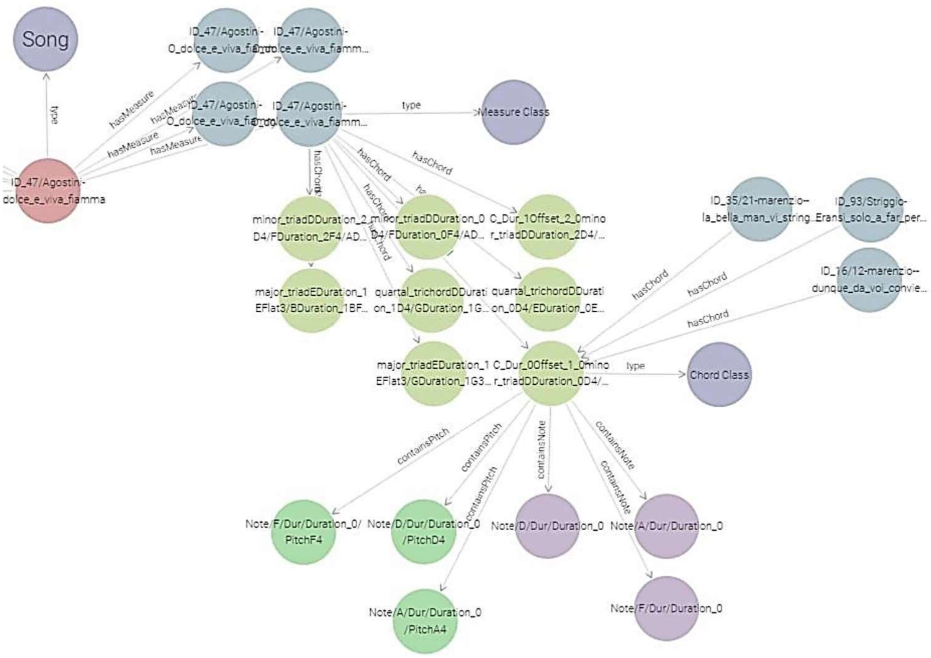

Figure 2. Node expansions

The formulas for each mentioned parameter are the following:

$$s_c = \frac{Nr_{identical_{chord}}}{Nr_{total\_chord}}$$

$$s_{ml} = 1 - |Len_{m1} - Len_{m2}| : |Len_{m1} - Len_{m2}| \in [0, 1]$$

$$Sr = Scos(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||} : \text{A, B – rhythm vectors}$$

$$Spd = Sj(A, B) = \frac{a\_11}{a\_11 + b\_01 + c\_10} : \text{A, B – binary vectors}$$

Integral-based method

Based on a melody integration approach [21], the aim is to examine the similarity between different measures through another mathematical approach by representing them as functions, and their resemblance is quantified by computing the area between two curves, both over a specified interval (Figure 3). This can be achieved by using the frequencies of the notes that compose a chord played during the measure.

$$f(x) = \sum_i \sin(2\pi * f_i * x)$$

Various techniques can be used to achieve results by integrating. These include changing how measures are expressed as functions by using a representation of a continuous wave, the individual absolute value of a function (Figure 4), or the mean of the frequencies, providing a different result for the same intervals on which the integration is performed. Considering how music sung with the voice can be perceived, one important assumption that can be made is that a chord is considered continuously played if it consists of a list of frequencies that remain unchanged over multiple time units. Conversely, if at least one frequency value changes, it is regarded as a new chord; therefore, the interval used for integration might be chosen based on different times computed using the measure divisions and chord durations.

The decision to examine the method in this manner came as a result of more case where the ratings obtained from the computations can be erroneous, indicating a resemblance between measurements that were very dissimilar.

The area between the functions obtained from measures is calculated as a sum of the integrals of the difference between the functions on each sub-interval, but with the previous condition described, having different integration intervals for f(x) and g(x), the absolute value could be used for each individual function.

$$Atotal = \sum_i \left| \int_{a_i}^{b_i} f_i(x) - \int_{c_i}^{d_i} g_i(x) \right|$$

Consequently, the area value might serve as an indicator of the resemblance between the two measures. The closer the result is to zero, the more comparable the functions expressing the measures are presumed to be. However, there are more cases in which this is not applicable due to multiple factors, such as a small value change of the primitive between interval limits, or close means computed from different frequencies.

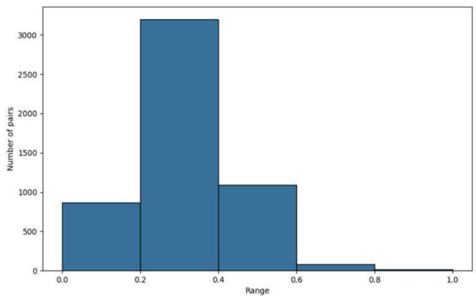

Figure 5. Distribution of similarity values

RESULTS

Initially, the training results of the Node2Vec model will be presented. To verify the reliability of these results, a manual analysis in collaboration with an experienced professional was required. Their expertise was crucial in confirming the correctness of the acquired values. Five pairs, denoted by P1, P2, P3, P4, and P5, were selected based on specific criteria for analysis.

- P1 - Songs written by the same author, with a high similarity value.
- P2 - Songs written by different authors, with a high similarity value.
- P3 - Songs written by the same author, with a medium similarity value.
- P4 - Songs written by different authors, with a low similarity value.
- P5 - Songs written by the same author, with a very low similarity value.

These conditions were chosen based on possible questions that might appear based on comparison cases, for example, exploring whether a high result is given only for music scores written by the same author or not, since those compositions might be more comparable because of a highly similar style particular to a composer, employed in numerous of his musical works. Table 1 shows the comparison values for the mentioned pairs.

Pair	First song	Second song	Value
P1	*Anima del cor mio*	*Amor, i' parto*	0.659
P2	*Deh dolce anima mia*	*L'aura che'l verde lauro*	0.525
P3	*Si dolce Amor mi fu quella cerasa*	*Tanto è la pena e sì grav'è'l dolore*	0.496
P4	*Occhi lucenti e belli*	*Piansi Donna per voi*	0.349
P5	*Cruda Amarilli*	*Dunque da voi convien*	0.233

Table 1. Similarity results for selected pairs

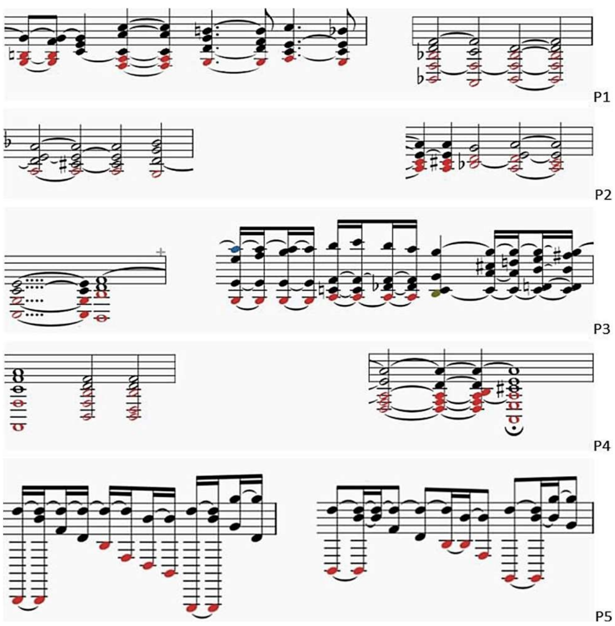

Figure 6. Extracted measure pairs

The main analyzing criteria during the verifications were the voices, the features of the song's parts, and the rhythm. Through the manual analysis, it was tried to obtain indicative marks to make a comparison between the results, noting that the output of the algorithm produced higher scores than expected. Despite the differences in numerical values, the ranking was still appropriate. This may be due to the fact that the model is more limited in terms of training data. When new individuals are added to the knowledge graph, the ratings for the same pairs may alter due to changes in the node network. Initially, considering this case, the findings can be promising, but there are uncertainties about the model's accuracy that arise after reviewing the most comparable song pairs. Especially in the case of pairs flagged as very similar (Table 2), the values can indicate a higher level of inaccuracy. In these instances, there are many differences in the structure of the musical scores to be considered as similar as the algorithm's output suggests. Due to this behavior, it cannot be said that the model is highly accurate.

First song	Second song	Value
Poi che di si vil foco	*Se per servirti*	0.894
Se per servirti	*Seguir' una ch'odia*	0.879
No.5. Se m'uccidi crudele	*No.6. Non posso più soffrire*	0.878
Ond'è'l lume gentil?	*Alma d'Amor gioiosa*	0.851
Poi che di si vil foco	*Seguir' una ch'odia*	0.844
Justorum animae	*In omnem teram*	0.840
Lex Domini immaculata	*Domine exaudi orationem meam*	0.821
Se per servirti	*No.6. Non posso più soffrire*	0.807
Seguir' una ch'odia	*No.6. Non posso più soffrire*	0.798
Se per servirti	*No.5. Se m'uccidi crudele*	0.790

Table 2. Highest similarity values

Each melody has a forward movement, with the notes being played from the beginning in a certain arrangement, which can be regarded as an attribute in the comparison of compositions, and this might also have an impact on the results. Inside the knowledge graph, the only elements that can bring a weight of order, since node properties are not evaluated by Node2Vec, might be represented by unique measures, numbered for each individual song. From the distribution of values presented in Figure 5, it can be observed that most pairs are marked as not very comparable.

The measures were also analyzed by manual comparison to identify the veracity of the values calculated by the methods described in the previous sections. The aim was to check for a consistent pattern, facilitating the formulation of significant conclusions based on the data.

Considering the values shown in Table 3, certain things can be generalized, starting with the level of accuracy in the case of each pair (Figure 6). When examining the integration- based method without incorporating frequency averages or the absolute value of functions, the results appear inaccurate; however, it can happen that the value reflects human analysis by chance.

Pair	Integral	Absolute value	Mean	Criteria	Node2 Vec
P1	0.032	3.453	2897.82	0.267	0.266
P2	0.098	1.290	495.08	0.559	0.731
P3	0.033	1.175	415.15	0.200	0.261
P4	0.262	4.903	3684.04	0.376	0.373
P5	0.019	0.432	368.50	0.551	0.699

Table 3. Similarity results for measures

The absolute and mean values can be accurate for P1 and P2. Even if there are still more situations where the value is misleading, it might be an improvement over the integration of the simple function. The disadvantage of using means is that a near mean value could be obtained from a wide range of frequencies, and in the case of using the absolute value of each function, there are different scenarios where the area resulting from the subtraction can be lower than expected. The values obtained using the criteria approach appear to be greater than anticipated, especially for some pairs, but they are more accurate, yet there are instances, such as P4, where the result is more erroneous. An interesting comparison could be made between Node2Vec and the criteria-based method, which has so far performed better compared to the other techniques.

The comparable similarity rating can lead to the belief that the model can offer accurate ratings (with exceptions for P2 and P5, where the values are significantly higher for N2V), yet by evaluating other pairs of measures, it is revealed that it is not a secure alternative for obtaining precise results.

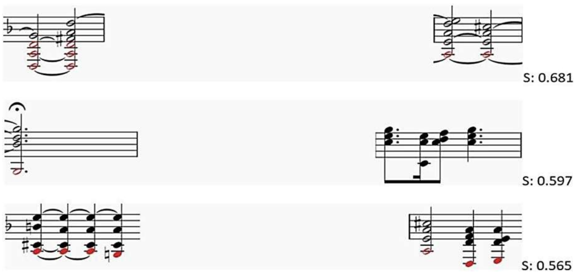

Figure 7. Node2Vec measure samples

As depicted in Figure 7, the measures are not as comparable as suggested by the model's output; therefore, the overall reliability is inconsistent. Initially, these issues seem to be more evident in the case of higher values; however, upon observing the second and third pair from the same figure, along with several others from the total set, it becomes apparent that the errors have a wider distribution.

CONCLUSIONS AND FUTURE WORK

This research explored methods for studying similarities between musical pieces in MusicXML format. Extracted features were used to populate a knowledge graph based on a new ontology, providing an organized representation of musical elements and relationships. The knowledge graph enhances music analysis by enabling complex reasoning, advanced querying, and better understanding of connections between musical components. RDF triples used as a method for structured representations offers a broader, more contextualized perspective on music analysis.

The comparative investigation of similarity ratings produced from embeddings, criteria-based and integral-based approaches has contributed to a better understanding of their use and robustness in the context of musical measure analysis. The criteria-based technique produces more reliable and consistent results, its strengths being represented by the ability to incorporate various aspects of musical measures, such as rhythm, pitch distribution, and the number of identical chords, but the conclusion about its accuracy is based on a limited number of manually verified pairs. Considering the total set of results, there might be even more inaccurate values, as in the case of P4. An attempt can be made to define other factors, using chord offsets to incorporate the idea of the order in which a song is played. This could have a large impact on the resemblance values obtained with this procedure. The integral-based approach presented more disadvantages, failing to capture the differences between the measures in an acceptable way; still, this technique can be further refined by applying different changes to the functions or by incorporating different details about the measures and chords, after a deeper analysis of its behavior. Regarding the performance of the Node2Vec model, more data can be provided to the network, thus influencing the links between individuals in the knowledge graph, leading to possible improvements.

The application of the methods outlined in this study produced additional data that could define future approaches on this topic, potentially leading to enhanced efficiency in the analysis of similar musical compositions. In the future, these techniques could also be used alongside machine learning models or integrated into their development for this purpose.

REFERENCES

1. Cuthbert, M. S., Ariza, C., Friedland, L. (2011). "Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit". In Proceedings of the International Symposium on Music Information Retrieval.
2. Cuthbert, M.S., Ariza, C. (2010). "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.". In Proceedings of the International Society for Music Information Retrieval.
3. Cuthbert, M. S., Hadley, B., Johnson, L., Reyes, C. (2012). "Interoperable Digital Musicology Research via music21 Web Applications.". From Joint CLARIN-D/DARIAH Workshop at Digital Humanities Conference Hamburg.
4. Rothstein, J. (1992). "MIDI: A Comprehensive Introduction." (2nd ed.). 1-12. A-R Editions.
5. *Introduction*. (2013, April 02). Retrieved on July 10, 2024, from https://www.musicxml.com/publications/makemusic-

recordare/commercial-applications/introduction.
6. Anders, T., Miranda, E. (2011). Constraint Programming Systems for Modeling Music Theories and Composition. ACM Computing Surveys. 43. 30:1-30:38. 10.1145/1978802.1978809.
7. Duriez, T., Brunton, S.L., Noack, B.R. (2017). Machine Learning Control (MLC). In: Machine Learning Control – Taming Nonlinear Dynamics and Turbulence. Fluid Mechanics and Its Applications, vol 116. Springer, Cham. https://doi.org/10.1007/978-3-319-40624-4_2
8. Anders, T., Inden, B. (2019). “Machine learning of symbolic compositional rules with genetic programming: dissonance treatment in Palestrina.” PeerJ Comput. Sci. 5:e244 DOI 10.7717/peerj-cs.244
9. Bizer, C., Heath, T., Berners-Lee, T. (2009). “Linked Data: The Story so Far.” International Journal on Semantic Web and Information Systems. 5. 1-22. DOI:10.4018/jswis.2009081901.
10. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F. (2007). „The Music Ontology.” Proceedings of the 8th International Conference on Music Information Retrieval. ISMIR 2007.
11. Poltronieri, A., Gangemi, A. (2022). The HaMSE Ontology: Using Semantic Technologies to support Music Representation Interoperability and Musicological Analysis. DOI:10.4230/OASIcs.MDK.2021.5. Retrieved on July 11, 2024, from: https://arxiv.org/abs/2202.05817
12. Poltronieri, A., Gangemi, A. (2023). „The Music Note Ontology.” Retrieved on July 11, 2024, from: https://ceur-ws.org/Vol-3011/pattern2.pdf
13. Chandrasekaran, B., Josephson, J., Benjamins, V. Richard. (1999). What Are Ontologies, and Why Do We Need Them?. Intelligent Systems and their Applications, IEEE. 14. 20 - 26. 10.1109/5254.747902.
14. Fazekas, G., Sandler, M. (2009). Novel methods in information management for advanced audio workflows. In proceedings of 12th International Conference on Digital Audio Effects. Como, Italy.
15. Gabbolini, G., Bridge, D. (2021). “An Interpretable Music Similarity Measure Based On Path Interestingnes.” arXiv:2108.01632v2 [cs.IR]
16. Maillot, P., Bobed, C. (2018). “Measuring structural similarity between RDF graphs.” 33rd Annual ACM Symposium on Applied Computing, Pau, France. 1960- 1967, ff10.1145/3167132.3167342ff. ffhal-01940449f
17. López-García, A., Martínez-Rodríguez, B., Liern, V. (2022). A Proposal to Compare the Similarity Between Musical Products. One More Step for Automated Plagiarism Detection? 10.1007/978-3-031-07015-0_16
18. Park, J., Kim, S., Shin, M. (2005). Music Plagiarism Detection Using Melody Databases. 684-693. 10.1007/11553939_98.
19. Prisco, R., Malandrino, D., Zaccagnino, G., Zaccagnino, R. (2017). A computational intelligence text-based detection system of music plagiarism. 519- 524. 10.1109/ICSAI.2017.8248347
20. Grover, A., Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. KDD: proceedings. International Conference on Knowledge Discovery & Data Mining. 855-864. 10.1145/2939672.2939754.
21. Manuel, E. (2008) A phenomenological approach to artificial jazz improvisation. University of Osnabruck
22. Laurson, M. 1996. PATCHWORK: A Visual Programming Language and some Musical Applications. Ph.D. thesis, Sibelius Academy, Helsinki.
23. Pachet, F. (1994). The MusES system: an environment for experimenting with knowledge representation techniques in tonal harmony. First Brazilian Symposium on Computer Music- SBC&M '94, August 3-4, Caxambu, Minas Gerais, Brazil, 195-201, 1994.
24. Roy, P., Pachet, F. 1997. Reifying Constraint Satisfaction in Smalltalk. Journal of Object-Oriented Programming 10, 4, 43–51.
25. Truchet, C., Assayag, G., Codognet, P. 2003. OMClouds, a heuristic solver for musical constraints. In MIC2003: The Fifth Metaheuristics International Conference. Kyoto, Japan.
26. Anders, T., Anagnostopoulou, C., Alcorn, M. (2004). Strasheela: Design and Usage of a Music Composition Environment Based on the Oz Programming Model. 277-291. 10.1007/978-3-540-31845-3_23