The Music Part Ontology (Extended Abstract)

Spyridon Kantarelis[1], Edmund Dervakos[1] and Giorgos Stamou[1]

[1]Artificial Intelligence and Learning Systems Laboratory, National Technical University of Athens, Heroon Polytechniou 9, Zografou, Attica, 157 80, Greece

Abstract

Symbolic representations of music play a vital role in the field of computer science, describing various musical elements such as notes, chord progressions, parts and structure, distinguishing them from audio formats. A plethora of user-generated symbolic music data lies on the web; however, for it to be valuable it necessitates to be processed and defined in a machine-readable way. This paper describes the creation of the Music Part Ontology (MPO), an ontology designed to formalize and analyze symbolic representations of music tracks based on their structural attributes using Description Logics and Semantic Web techniques and showcases its practical application in the Music Information Retrieval (MIR) domain.

Keywords

Ontology Engineering, Description Logics, Semantic Web, Music Information Retrieval, Music Structure

1. Introduction

Data curation and management hold significant importance in the field of Computer Science. The advancements in Artificial Intelligence (AI) have given rise to many tools which rely on well-defined data and metadata in order to achieve high quality and trust-worthy results. Particularly, when dealing with symbolic music data, curation and standardization become necessary as a large amount of it is sourced from user-generated data on the web[1]. This raw data needs to be formalized in order to be effectively utilized in Muisc Information Retrieval (MIR) tasks. More specifically, our ontology specializes in converting raw text data into a graph representation.

Various ontologies have been introduced dealing with music-related data and can be categorized into two main directions; one direction focuses on the description of concepts related to music production and performance, audio terms and general low-level music features [1, 2, 3], while the other one concentrates on high-level music concepts derived from music notation, theory and harmony notions [4, 5, 6] Although these ontologies facilitate real-world applications such as managing music collections, transcribing handwritten music and formalizing and inferring music knowledge, none of them capitalize on the structure of a music track and the sequential nature of its fundamental components (rhythm, melody and harmony).

DL 2023: 36th International Workshop on Description Logics, September 2–4, 2023, Rhodes, Greece

spyroskanta@ails.ece.ntua.gr (S. Kantarelis); eddiedervakos@islab.ntua.gr (E. Dervakos); gstam@cs.ntua.gr (G. Stamou)

0000-0001-8034-8047 (S. Kantarelis); 0000-0001-7838-3919 (E. Dervakos); 0000-0003-1210-9874 (G. Stamou)

[1]http://www.ultimate-guitar.com, http://www.guitartabs.cc, http://www.e-chords.com

In the proposed approach, we develop and introduce the Music Part Ontology (MPO)[2]; we define a set of concepts and relationships relevant to the structural attributes and sequential aspects of contemporary western music. For example, the ontology defines terms such as *Intro, Verse, ChordProgression*, and a music track can be assigned a sequence of these terms. Furthermore, we demonstrate their ability to be interlinked, utilizing Semantic Web technologies. For example, by defining the chord concept, the MPO ontology can be linked with the Functional Harmony Ontology (FHO)[3], the Music Theory Ontology (MTO) and the Chord Ontology[4]. Finally, we showcase its capabilities by gathering user-generated symbolic music data and converting them into a well-defined music knowledge base where we apply SPARQL queries to approach the genre classification MIR task. More specifically, we demonstrate how the ontology can be used to facilitate complex music related queries, such as "Retrieve all songs that have a ii-V-I progression in the intro".

2. The Music Part Ontology

The MPO was developed to formalize one the most common representations of symbolic music data which is the text representation consisting of parts and chord progressions over lyrics and is ubiquitous on the web. We introduce four main Classes: *Track, Part, ChordProgression* and *Chord* which are connected through the properties *hasPart, hasChordProgression* and *hasProgChord* to describe the structure of a music track. In order to represent the sequential aspect of a music track we chain parts, chord progressions and chords with *hasNextPart, hasNextChordProgression* and *hasNextChord* functional properties respectively. The format of MPO is depicted in Figure 1. Furthermore, for each chain property *hasNextX*, we created the ancestral transivite property *isFollowedbyX*. This enables us to infer that for *a, b, c* instances, if *a hasNextX b* and *b hasNextX c*, then *a isFollowedbyX c*, as introduced in [7].

Musical parts are hard to define, as they can vary across genres and cultures. For example, in electronic dance music, the parts such as "build", "drop" are defined mainly based on the dynamics and timbral characteristics of the music. In other genres, such as pop and rock, different parts (e.g. verse, chorus) are often characterized by difference in harmony, such as different chord progressions. We ended up defining eight parts as subclasses of the class *Part*: *Intro, Verse, Chorus, Outro, Bridge, Instrumental, Interlude* and *Solo*, as these are the most common parts used in prior works [8, 9]. Each instance of a subclass is constructed by defining each first and last chord using the properties *hasFirstPartChord* and *hasLastPartChord.* We also created the *hasPartChord* property to link every part with all their chords. Additionally, for each part we defined the properties *hasVerse, hasChorus,* etc as subproperties of *hasPart.*

Chord progressions are the foundations of a music track's harmony. Common chord progressions in western music contain usually three to eight chords. In order to create instances of the *ChordProgression* class we define its first and last chord using the properties *hasFirstProgChord* and *hasLastProgChord.*

There are multiple ways to represent music chords. We utilize the vocabulary from the FHO

[2] http://purl.org/ontology/mpo
[3] https://purl.org/ontology/fho
[4] https://purl.org/ontology/chord/

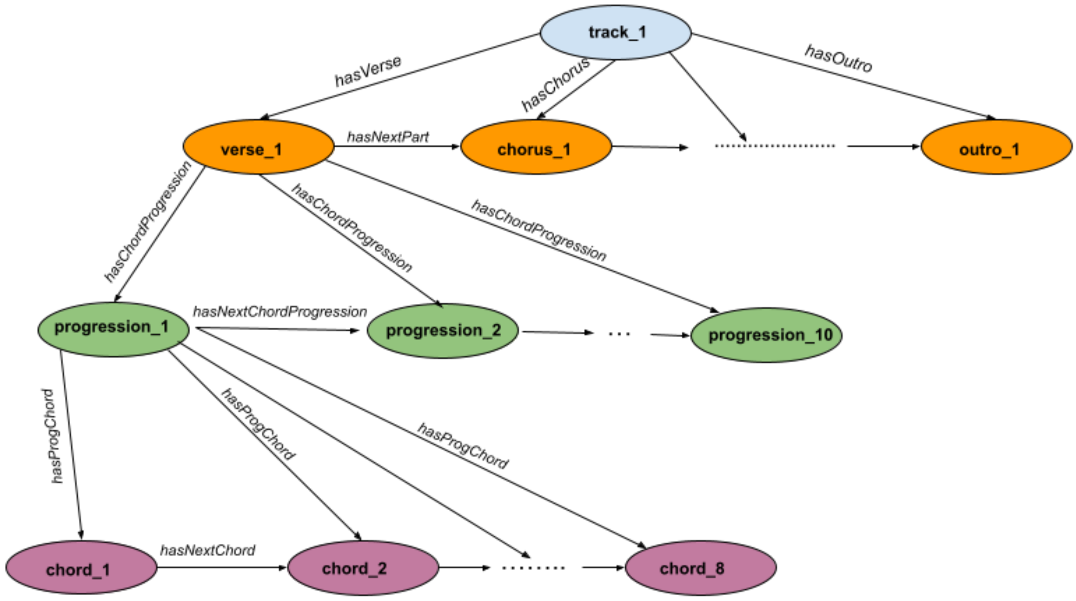

Figure 1: The Music Part Ontology format

by linking our *Chord* class with it, which is also interlinked with concepts from the Music Theory Ontology and the Chord Ontology.

3. Demonstration

In order to evaluate the practical application of our proposed ontology, we set up an experiment. First, we gathered user-generated symbolic music data from **ultimate-guitar** in text form. Then, we curated the data: we transformed the chords to match the proposed chord vocabulary and also filtered the different part names to correspond to the proposed subclasses of the *Part* class (e.g. coda, ending and outro align with *Outro* class). In addition, we retrieved the music genres of the gathered tracks using the Spotify Web API [5] and created the *Genre* class and its subclasses (*Rock, Pop, Jazz,* etc) and the property *hasGenre.* After that, all the music tracks were converted into an RDF graph using the RDFLib Python package[6]. Lastly, we set up a semantic repository on GraphDB[7] and loaded the MPO, the FHO and the RDF graphs of the music tracks. We ended up with 250 music tracks of five genres: (*Rock, Pop, Jazz, Blues* and *Reggae*).

Having ontologies that define music theoretical notions, and large collections of music that are semantically characterized, we can perform complex SPARQL queries, that can be especially useful for music information retrieval, as a way to find patterns based on chord progressions. For example, we can retrieve all *ii-V-I* progressions (that are regularly used in Jazz) and see

[5] https://developer.spotify.com/documentation/web-api
[6] https://rdflib.readthedocs.io/en/stable/
[7] https://graphdb.ontotext.com/

their distribution between different genres and therefore could enhance the performance of a genre classification model [10, 11], e.g. by deploying these results as a complementary input. Table 1 shows the results of this query. Table 2 shows the results of a query about the *Chorus* distribution between genres, a much simpler query.

On that account, it is worth mentioning that the structural and sequential attributes of a music track have the potential to boost the effectiveness of a classification model. This is evident when we observe that by using both complex and simple queries, we can obtain further insights into the distinctions and resemblances among various music genres.

On our GitHub[8], we provide some SPARQL queries examples, some RDF files and text files of music tracks and the python script used to convert text to RDF.

Table 1
ii-V-I Distribution

Genre	ii-V-I Distribution
Rock	1%
Jazz	**2.3%**
Pop	1.56%
Blues	1.38%
Reggae	2.11%

Table 2
Chorus Distribution

Genre	Chorus Distribution
Rock	21.9%
Jazz	27.1%
Pop	24.7%
Blues	22.3%
Reggae	**33.3%**

4. Conclusion

In this paper we introduced the Music Part Ontology. We showed its capability for formalizing and converting raw symbolic music data into a graph representation. Our ontology can be easily interlinked with other ontologies of the MIR domain. In addition, we showcased its practical application by performing SPARQL queries that can be exploited to enhance the performance of models on MIR tasks, such as genre classification. We plan to further extend our ontology to include more concepts about the structure of a music track and explore ways to retrieve and define common chord progressions through reasoning.

Acknowledgments

The authors would like to thank researchers Konstantinos Thomas and Vassilis Lyberatos of the Artificial Intelligence and Learning Systems Laboratory for their contribution on gathering the data used in section 3.

References

[1] Y. Raimond, S. A. Abdallah, M. B. Sandler, F. Giasson, The music ontology., in: ISMIR, volume 2007, Citeseer, 2007, p. 8th.

[8] https://github.com/spyroskantarelis/MusicPartOntology

[2] S. Song, M. Kim, S. Rho, E. Hwang, Music ontology for mood and situation reasoning to support music retrieval and recommendation, in: 2009 Third International Conference on Digital Society, 2009, pp. 304–309.
[3] B. Fields, K. Page, D. De Roure, T. Crawford, The segment ontology: Bridging music-generic and domain-specific, 2011, pp. 1–6. doi:10.1109/ICME.2011.6012204.
[4] S. M. Rashid, D. De Roure, D. L. McGuinness, A music theory ontology, in: Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music, SAAM '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 6–14. URL: https://doi.org/10.1145/3243907.3243913. doi:10.1145/3243907.3243913.
[5] S. S.-s. Cherfi, C. Guillotel, F. Hamdi, P. Rigaux, N. Travers, Ontology-based annotation of music scores, in: Proceedings of the Knowledge Capture Conference, K-CAP 2017, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3148011.3148038. doi:10.1145/3148011.3148038.
[6] S. Kantarelis, E. Dervakos, N. Kotsani, G. Stamou, Functional harmony ontology: Musical harmony analysis with description logics, Journal of Web Semantics 75 (2023) 100754. URL: https://www.sciencedirect.com/science/article/pii/S1570826822000385. doi:https://doi.org/10.1016/j.websem.2022.100754.
[7] N. Drummond, A. L. Rector, R. Stevens, G. Moulton, M. Horridge, H. Wang, J. Seidenberg, Putting owl in order: Patterns for sequences in owl., in: OWLED, 2006.
[8] J.-C. Wang, Y.-N. Hung, J. B. Smith, To catch a chorus, verse, intro, or anything else: Analyzing a song with structural functions, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 416–420.
[9] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, J. S. Downie, Design and creation of a large-scale database of structural annotations., in: ISMIR, volume 11, Miami, FL, 2011, pp. 555–560.
[10] N. Ndou, R. Ajoodha, A. Jadhav, Music genre classification: A review of deep-learning and traditional machine-learning approaches, in: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE, 2021, pp. 1–6.
[11] E. Dervakos, N. Kotsani, G. Stamou, Genre recognition from symbolic music with cnns, in: Intern. Conf. on Comp. Intelligence in Music, Sound, Art and Design (Part of EvoStar), Springer, 2021.