SANGEET: A XML based Open Dataset for Research in Hindustani Sangeet

Chandan Misra and Swarup Chattopadhyay *

School of Computer Science & Engineering, XIM University
chandan@xim.edu.in, swarupc@xim.edu.in

Abstract. It is very important to access a rich music dataset that is useful in a wide variety of applications. Currently, available datasets are mostly focused on storing vocal or instrumental recording data and ignoring the requirement of its visual representation and retrieval. This paper attempts to build an XML-based public dataset, called SANGEET, that stores comprehensive information of *Hindustani Sangeet* (North Indian Classical Music) compositions written by famous musicologist *Pt. Vishnu Narayan Bhatkhande*. SANGEET preserves all the required information of any given composition including metadata, structural, notational, rhythmic, and melodic information in a standardized way for easy and efficient storage and extraction of musical information. The dataset is intended to provide the ground truth information for music information research tasks, thereby supporting several data driven analysis from a machine learning perspective. We present the usefulness of the dataset by demonstrating its application on music information retrieval using XQuery, visualization through *Omenad* rendering system. Finally, we propose approaches to transform the dataset for performing statistical and machine learning tasks for a better understanding of *Hindustani Sangeet*. The dataset can be found at `https://github.com/cmisra/Sangeet`.

Keywords: Hindustani Sangeet, North Indian Classical Music, XML, Music Dataset, Classification, XQuery, Music Rendition

1 Introduction

Having access to free, well-maintained databases of music is a crucial resource for researchers. In the case of Indian Classical Music, this is also true since it has been shown to be important for high-quality research in music information retrieval (MIR) [8,11,21,15] and musicological analysis using machine learning [23,22,18], deep learning [12,20,16,19], etc. Several high-quality datasets, [1] and [2] for example, for research in MIR and computational musicology can be found in the published literature.

Although audio recording-based music corpora are essential in certain types of music applications, studies of existing literature reveal a dearth of substantial research related to the varied domains of *Sound and Music Computing* (SMC), especially in the design and development of interfaces for expressing Indic Music on an electronic medium. One of the domains that interest us is the creation of an Indian music environment through the transcription and rendering of an Indic music piece using Indic notation systems and Indic language script. The ability to compose music electronically entirely in an Indian music environment necessitates the emergence of research in different domains in SMC. This requires a musicological analysis of the grammar and structure of the music sheets presently in use and the consequent development of musical fonts and rendering engines ([5] for staff notation for example). Needless to say, such endeavors would motivate the research community to create models for Indic music notation systems and their language bases [14] and provide ample opportunities to work in building interfaces for music expressions on an electronic medium.

The mere enabling of music practitioners in composing music electronically is insufficient unless we have tools to exchange such musical information seamlessly across applications. Consequently, this establishes the need for the development of a common music exchange format to communicate music independent of any genre, notation system, language script, and music sheet structure [13]. XML-based formats for exchanging musical information have existed for quite some time [13,10,7] and are being adopted by extremely robust and popular notation software like *Finale* [3], *Sibelius* [6], *MuseScore* etc. Additionally, the ability to store musical information in XML solves the problem of archiving our historic musical art form as an electronic database.

One of the most authentic sources of Hindustani Sangeet is the compositions published in the book *Hindusthani Sangeet Paddhati-Kramik Pustak Malika* which comprises of approximately 1900 compositions belonging to North Indian Classical Music penned by *Pt. Vishnu Narayan Bhatkhande* (1860 - 1936). In order to reach a greater number of music students and scholars, the first volume of *Kramik Pustak Malika* has been translated to *Hindi* language in 1953 by prominent music scholar *Dr. Laxminarayan Garg*. This paper introduces SANGEET, arguably the first XML-based music corpora that try to capture comprehensive musical information contained in these rich music sources to apply in various music applications like music transcription, visualization, MIR, computational musicology, etc. We begin the preparation of the dataset with the second volume of *Kramik Pustal Malika* book series and our objective is to store compositions of different genres in a carefully crafted XML database to preserve comprehensive musical information in a single format. This will provide the users to obtain a standard framework for efficient and easy access to the dataset that can be easily transformed to apply to various applications. We refer to three music applications related to visualization, MIR, and machine learning in support of the coverage, quality, and accessibility of SANGEET.

2   The Organization and Access of SANGEET

Pt. Vishnu Narayan Bhatkhande is the pioneer for providing a comprehensive theoretical foundation of Hindustani Sangeet in a published form in his six-volume book series

titled *Hindustani Sangeet Paddhati, Kramik Pustak Malika* written in *Marathi* language in 1920. His book contains a comprehensive description of music symbols for realizing musical components including notes (Svar), time signatures (Lay), beats (Taal), ornaments (Alankar) etc. The dataset created in the current work has been taken from the Hindi translation of the second volume of the series. The second volume of the book series contains a total of 319 compositions belonging to 10 different raags. The present work takes these written compositions as a source of musical information to create the database for Hindustani Sangeet to be used in various applications.

We have taken 116 compositions of the three highest frequent raags i.e. raag *Bhairav* (42), *Todi* (39), and *Poorvi* (35) respectively, from the entire collection of 319 compositions for performing our experimental analysis. Eventually, the entire collection of compositions from all six volumes will be preserved in the dataset for applications related to music information retrieval, music-sheet visualization, etc.

The dataset consists of a number of XML documents that is equal to the number of compositions in the dataset i.e. each XML document represents a single composition of the dataset. The XML documents are equipped with meaningful tags to store all the necessary musical information for the compositions. The format of the XML files is validated against a schema definition document so that the format of the dataset or compositions are preserved. The schema definition document is an XML Schema Definition (XSD) file against which each XML document is checked and validated for legal elements and attributes. The XSD consists of four parts namely *info*, *taal*, *raag*, and *sheet* responsible for storing metadata, rhythmic, melodic, structural, and notational information in the XML files.

The metadata linked to the musical composition is represented by the *info* portion. It contains information on the catalog, the genre, and the notational system as shown in Listing 1.1 describing the first composition of the second volume of the book.

```
1  <INFO>
2    <TITLE>Composition 1 Volume 2 Kramik Pustak Malika</TITLE>
3    <AUTHOR>Pt. Vishnu Narayan Bhatkhande</AUTHOR>
4    <NOTATION_SYSTEM>Bhatkhande</NOTATION_SYSTEM>
5    <DATE_TIME>1923</DATE_TIME>
6    <GENRE>Hindusthani Sangeet</GENRE>
7    <ADDITIONAL>
8      <ENTRY>http://ndl.iitkgp.ac.in/document/
       R2pPWGRxdkRWWnlvOVdPYzdzaWpTV0pYYTFIT0VnNTB6V1dnR1dJVW1kUT0</ENTRY>
9    </ADDITIONAL>
10 </INFO>
```

Listing 1.1: Info Part of XML file depicting metadata

The rhythmic foundation of Indian music is provided by *taal*. Indic music has nearly hundreds of Taals, each with its own specific composition that includes the name, *Bibhaga* or measure, *Maatra* or the number of beats, *Avartana* or the number of cycles per line, etc. Additionally, Taal has two designated beat indices, known as *Taali* and *Khali*, to signify stressed or unstressed strokes in addition to a specific beat pattern to uniquely identify a Taal. These patterns, which are required to portray the Taal graphically or as a music sheet accompanied by an Indian percussionist, have been illustrated as a series of numbers (seen in Listing 1.2). Additionally, the regular expression specifies the expression for a beat pattern, making it easier to query the Taal's structure.

```
1  <TAAL>
```

```
 2    <TAAL_NAME>Tritaal</TAAL_NAME>
 3    <BIBHAGA>4</BIBHAGA>
 4    <MAATRA>16</MAATRA>
 5    <AVARTANA>1</AVARTANA>
 6    <BEAT_PATTERN>4-4-4-4</BEAT_PATTERN>
 7    <ALTERNATE_BEAT_PATTERN>NA</ALTERNATE_BEAT_PATTERN>
 8    <TAALI_COUNT>3</TAALI_COUNT>
 9    <KHALI_COUNT>1</KHALI_COUNT>
10    <TAALI_INDEX>1-5-13</TAALI_INDEX>
11    <KHALI_INDEX>9</KHALI_INDEX>
12  </TAAL>
```

Listing 1.2: Taal Part of XML file depicting Taal and its sub-components

Raag provides the melodic framework to Hindustani Sangeet and each raag can be identified by characteristics like *Arohana* and *Avarohana*, which are ascending or descending movements made up of a series of notes, *Vadi* and *Samvadi*, which are consonant and dissonant notes, and classification forms like *Pakad* and *Jaati*. These characteristics are note sequences and have been encoded using *Ome Swarlipi* [4], the same rendition we use for storing notes in our dataset.

```
 1  <RAAG>
 2    <RAAG_NAME>Yaman</RAAG_NAME>
 3    <THAAT>Kalyan</THAAT>
 4    <AROHANA>n-r-g-M-d-n-su</AROHANA>
 5    <AVAROHANA>su-n-d-p-M-g-r-s</AVAROHANA>
 6    <VADI>g</VADI>
 7    <SAMVADI>n</SAMVADI>
 8    <JAATI>Sampoorna</JAATI>
 9    <PAKAD>nlrgr-s-pMg-su</PAKAD>
10  </RAAG>
```

Listing 1.3: Raag Part of XML file depicting Raag and its sub-components

Sheet, which is based on the 2D matrix model *Swaralipi* [14], specifies the layout of the music sheet and the placement of the notation symbols. As a result, it replicates the entirety of the contents as a rectangular row-column arrangement. Even though we haven't yet transcribed the beat markings and lyrics, the model has the provision to include them in the future. The format cleverly transforms row and column models into helpful tags that make it easier to develop various applications, such as real-time note playback, producing music sheets, and retrieving score data. For example, part of the first line of the original composition (shown in Figure 1a) has been converted into the sheet part (shown in Figure 1b and 1c).

3   Applications of the Dataset

Visualization of Music-sheets: One of the primary applications of any music dataset is to visualize it or render it using a notation system in which it is preserved. We have encountered several difficulties in visualizing the composition in the Bhatkhande notation system since there is no standard font system for rendering Bhatkhande music symbols in any language script. The closest rendition we have found is the *Ome Swarlipi* [4] system which is a compact version of the Bhatkhande notation system and easy to use. In order to visualize in HTML format, the system provides the necessary styling information to render it in *Devanagari* script. Therefore the pre-processing step for this

(a)

```
1  <SHEET>
2    <TOTAL_LINE></TOTAL_LINE>
3    <LINES>
4      <LINE INDEX="1">
5        <ROW INDEX="1">
6          <COL INDEX="1">
7            <NOTE_COUNT>1</NOTE_COUNT>
8            <CONTENT>n</CONTENT>
9          </COL>
10         <COL INDEX="2">
11           <NOTE_COUNT>1</NOTE_COUNT>
12           <CONTENT>d</CONTENT>
13         </COL>
14         <COL INDEX="3">
15           <NOTE_COUNT>1</NOTE_COUNT>
16           <CONTENT>p</CONTENT>
17         </COL>
18         <COL INDEX="4">
19           <NOTE_COUNT>1</NOTE_COUNT>
20           <CONTENT>M</CONTENT>
21         </COL>
22       </ROW>
23       ......
```

```
1  <LINE INDEX="2">
2      <ROW INDEX="1">
3        <COL INDEX="1">
4          <NOTE_COUNT>1</NOTE_COUNT>
5          <CONTENT>g</CONTENT>
6        </COL>
7        <COL INDEX="2">
8          <NOTE_COUNT>1</NOTE_COUNT>
9          <CONTENT>M</CONTENT>
10       </COL>
11       <COL INDEX="3">
12         <NOTE_COUNT>1</NOTE_COUNT>
13         <CONTENT>p</CONTENT>
14       </COL>
15       <COL INDEX="4">
16         <NOTE_COUNT>1</NOTE_COUNT>
17         <CONTENT>M</CONTENT>
18       </COL>
19     </ROW>
20   </LINE>
21   </LINES>
22 </SHEET>
```

(b)   (c)

Fig. 1: Sheet part of the XML file (b) and (c) depicting part of the original music sheet (a).

application is a converter that takes an XML file as a standalone composition and transforms it into equivalent HTML with the *Ome Swarlipi* rendition of the score. The source code of the converter has been given in the online repository link and the corresponding rendition is shown in Figure 2.

Query and Retrieval of Musical Information: This is the application where we can appreciate the power of XML as a means to build the music dataset. XML has brought with it a number of tools and technologies to efficiently process the information contained inside it. For the present application, we have used two tools, namely *XPath* and *XQuery*. XPath, the XML Path Language, uses path expressions to parse through the elements and attributes of an XML document and select node elements to extract the contents inside it. This language is also used in another query language XQuery to query an XML database and retrieve required information from it much like the SQL that does the same on a relational database.

The *preprocessing* stage for this application is to create an XML database created from the XML documents. We have used *BaseX* database engine to create the database from our dataset and XQuery to efficiently and easily perform complex queries and retrieve information from it and therefore, can be extremely useful for data-intensive complex web applications. This also provides a single-point query and retrieval system, as opposed to the current search and retrieval platforms [9,17] used for querying and
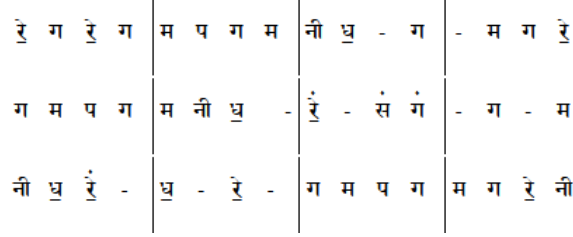
Fig. 2: Music-sheet web visualization using Ome Swarlipi

```
1  (: List of compositions having Meend :)
2  for $songs in collection ("
      Bhatkhande-Database")//swarlipi
3  let $title := $songs/INFO/TITLE/text()
4  let $contents := $songs/SHEET/LINES/
      LINE/ROW/COL/CONTENT/text()
5  let $notes := (for $song in $songs
6  return $song/SHEET/LINES/LINE/ROW/COL/
      CONTENT/text())
7  return if (contains(string-join($notes
      , ""),"q")) then
8  $title
```

(a)

```
1  (: List of compositions having a
      particular Arohana subsequence :)
2  for $songs in collection ("
      Bhatkhande-Database")//swarlipi
3  let $title := $songs/INFO/TITLE/text()
4  let $aroha := $songs/RAAG/AROHANA/text()
5  return if (contains($aroha, "s-R-g"))
         then
6  $title
```

(b)

```
1  (:  Note frequency distribution of each composition :)
2  for $song in collection("Bhatkhande-Database")//swarlipi
3  let $raag := $song/RAAG/RAAG_NAME/text()
4  let $contents := $song/SHEET/LINES/LINE/ROW/COL/CONTENT/text()
5  let $joined_str := string-join(data($contents), ',')
6  let $joined_str := replace($joined_str, "<sup>|</sup>|@|u|l|\)|\(|-|,|\s+", "")
7  let $notes := (115,82,114,71,103,109,77,112,68,100,78,110)
8  let $code_points := string-to-codepoints($joined_str)
9  let $result := (for $i in $notes
10    return count(index-of($code_points, $i))  )
11 let $result := normalize-space(string-join($result, ","))
12 return $result
```

(c)

Fig. 3: XQuery to retrieve the (a) list of compositions having Meend, (b) List of compositions having a particular *Arohana* subsequence and (c) Note frequency distribution of each composition

browsing musical data. Figure 3 provides a few interesting and complex queries that satisfy the fine-grained information needs of the user. For example query 3c can be used to generate dataset for raag classification as described in the following section.

Raag prediction through Machine Learning: This application refers to the musicological analysis of various musical components present in Hindustani Sangeet. It covers statistical and structural analysis, data mining, and inference using machine learning and deep learning techniques. As an example of the application, we apply machine learning techniques on the dataset for the task of raag prediction. The *preprocessing*

Accuracy Score of Classification Models				
Logistic Regression	K-Nearest-Neighbors (KNN)			Decision Tree
	$k = 3$	$k = 5$	$k = 7$	
0.9143	0.9714	0.9428	0.9428	0.9714

Table 1: Performance measure of Logistic Regression, K-Nearest Neighbors with varying values of k, and Decision Tree. The dataset is divided into 70:30 as training and test set to calculate the accuracy score of different classification models.

step for raag prediction is to convert the XML dataset into a tabular data-frame containing a number of features and a target variable. For raag prediction, we take features as the frequencies of individual notes and the corresponding raag as a target variable for any composition. Instead of taking the note-frequency distribution of 36 notes for a composition spanning across three octaves, we merge the notes to obtain the frequency distribution of 12 notes. Since, the positions of the notes of the *Arohana* and *Avarohana* of any particular composition in different octaves do not affect the raag of the composition, we map corresponding notes of three octaves and make a sum of frequencies of corresponding notes to obtain 12 note-frequency distribution (can be obtained from 3c given in GitHub). Table 1 shows the measure of performance of different machine learning techniques for raag prediction for our dataset. We have transformed our dataset into a three-class classification problem by taking the three most frequent raags i.e. *Bhairav*, *Todi*, and *Poorvi*, and applied the different classification models to generate the accuracy scores. Since each classifier examined shows high accuracy score the dataset can be considered as a robust dataset for raag classification. Table 1 shows that KNN with $k = 3$ and decision tree classifier gives better accuracy scores than the logistic regression model.

4 Conclusions and Future Works

This paper presents SANGEET, a Hindustani Sangeet dataset based on XML to provide easy and efficient access to a music corpora to perform various applications including music visualization, MIR, and Raag prediction using machine learning techniques. Backed by a robust music-sheet framework and a structured XSD, SANGEET provides a comprehensive repository for rich musical information to be shared seamlessly across applications. We have shown that SANGEET is quite efficient for accessing and transforming musical data into a format suitable for various musical applications. Our future objective is to extend SANGEET with the compositions of Bhatkhande's other five volumes of *Kramik Pustak Malika* and update the structure of the XML dataset with taal markings and lyric information. This will provide better music-sheet rendition and richer queries to fulfill the user's information needs.

References

1. Annotated compmusic datasets. https://compmusic.upf.edu/datasets, accessed: 2022-05-6

2. Dunya. `https://dunya.compmusic.upf.edu/`, accessed: 2022-05-6
3. Finale—music notation software that lets you create your way. `https://www.finalemusic.com/`, accessed: 2022-05-6
4. Fonts for writing indian music - omenad fonts. `https://omenad.github.io/fonts/`, accessed: 2022-05-6
5. Lilypond... music notation for everyone. `http://lilypond.org/`, accessed: 2022-02-23
6. Music notation software - sibelius - avid. `https://www.avid.com/sibelius`, accessed: 2022-05-6
7. Baggi, D., Haus, G.: Ieee 1599: Music encoding and interaction. Computer 42(3), 84–87 (2009)
8. Chithra, S., Sinith, M., Gayathri, A.: Music information retrieval for polyphonic signals using hidden markov model. Procedia Computer Science 46, 381–387 (2015)
9. Ghosh, S., Dasgupta, A., Mukhopadhyay, D., Datta, D.: tagoreweb: The complete works of rabindranath tagore. `http://tagoreweb.in/` (2020), accessed: 2022-05-6
10. Good, M., et al.: Musicxml: An internet-friendly format for sheet music. In: XML Conference and Expo. pp. 3–4. Citeseer (2001)
11. Kirthika, P., Chattamvelli, R.: A review of raga based music classification and music information retrieval (mir). In: 2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA). pp. 1–5. IEEE (2012)
12. Madhusudhan, S.T., Chowdhary, G.: Deepsrgm-sequence classification and ranking in indian classical music with deep learning. In: Proceedings of the 20th International Society for Music Information Retrieval Conference. pp. 533–540 (2019)
13. Misra, C.: Sangeetxml: An xml format for score retrieval for indic music. In: ACM Multimedia Asia, pp. 1–5 (2021)
14. Misra, C., Chakraborty, T., Basu, A., Bhattacharya, B.: Swaralipi: A framework for transcribing and rendering indic music sheet (2016)
15. Murthy, Y.S., Koolagudi, S.G.: Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. ACM Computing Surveys (CSUR) 51(3), 1–46 (2018)
16. Nag, S., Basu, M., Sanyal, S., Banerjee, A., Ghosh, D.: On the application of deep learning and multifractal techniques to classify emotions and instruments using indian classical music. Physica A: Statistical Mechanics and Its Applications 597, 127261 (2022)
17. Society for Natural Language Technology Research, G.o.W.B.: rabindra-rachanabali. `https://rabindra-rachanabali.nltr.org` (1905), accessed: 2022-05-6
18. Patel, E., Chauhan, S.: Raag detection in music using supervised machine learning approach. International Journal of Advanced Technology and Engineering Exploration 4(29), 58 (2017)
19. Pendyala, V.S., Yadav, N., Kulkarni, C., Vadlamudi, L.: Towards building a deep learning based automated indian classical music tutor for the masses. Systems and Soft Computing 4, 200042 (2022)
20. Sharma, A.K., Aggarwal, G., Bhardwaj, S., Chakrabarti, P., Chakrabarti, T., Abawajy, J.H., Bhattacharyya, S., Mishra, R., Das, A., Mahdin, H.: Classification of indian classical music with time-series matching deep learning approach. IEEE Access 9, 102041–102052 (2021)
21. Sridhar, R., Geetha, T.: Raga identification of carnatic music for music information retrieval. International Journal of recent trends in Engineering 1(1), 571 (2009)
22. Sridharan, A., Moh, M., Moh, T.S.: Similarity estimation for classical indian music. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA). pp. 814–819. IEEE (2018)
23. Ujlambkar, A.M., Attar, V.Z.: Mood classification of indian popular music. In: Proceedings of the CUBE international information technology conference. pp. 278–283 (2012)