Music Retrieval Using Onomatopoeic Query

Kenji Ishihara, Fuminori Kimura and Akira Maeda

***Abstract*—We propose a music retrieval method that uses onomatopoeia query to enable easy and intuitive retrieval of music even for users without musical knowledge. When we use a music retrieval system, we usually need some information on the music, such as artist names or lyrics. However, if we do not know such information, it is difficult to retrieve the music, especially instrumental music. In the proposed method, we create an index of changes in note length and pitch in all of the music to be retrieved in advance. When searching, the proposed system extracts the change in note length and pitch from the user's query. The extracted information is used to calculate the similarity of a musical piece and the query by using a dynamic programming matching algorithm. The proposed system ranks a musical piece on the basis of similarity. The music format of the retrieval target is MIDI. In this paper, the result of retrieval experiments with the proposed method is shown.**

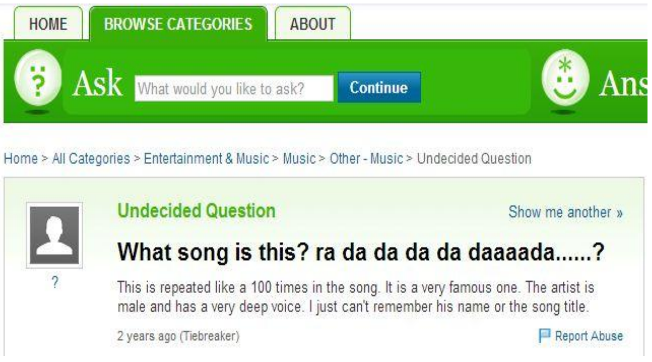
Keywords : Music Retrieval, Onomatopoeia, Indexing, MIDI

I. Introduction

Many people use music retrieval systems for various reasons. Recently, there have been various music retrieval systems that use a variety of queries. These systems often require users to input information on music, such as the metadata of the music, or to hum in order to search. However, users cannot retrieve music by using a keyword retrieval system with metadata if they do not have information on the music. Additionally, the system is not suitable for searching for music we hear out and about in the city. In addition, we have difficulty retrieving instrumental music that has no lyrics, unlike songs. Some voice search systems [1-6] are available with precise search that uses the pitch of music, but we some people sometimes hesitate to use the system in public places. In addition, it is not suitable for tone-deaf users.

We propose a music retrieval method in which a user enters the onomatopoeia as a query. Even if the user has a vague memory of the music, it can be entered instinctively. Generally, onomatopoeia expresses sound effects and animal sounds with words, for example, bow-wow, bong, and so on. However, we define onomatopoeia as if humming represented by character strings. It is common for a user to express a melody by using characters such as "la" and "daaaa", on bulletin board systems (BBSs) or in conversation in Japan, English-speaking countries, and so on. Hyphens are used to represent the length of the note in Japanese onomatopoeia, and users represent the difference in length by the number of consecutive hyphens. However, English-speakers represent the difference in the length by repeating the same characters such as "woh" or "daaaa" (Figure 1).

Kenji Ishihara is with the Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan (corresponding author to provide e-mail: is003085@ed.ritsumei.ac.jp).
Fuminori Kimura is with the Kinugasa Research Organization, Ritsumeikan University, Kyoto, Japan (e-mail: fkimura@is.ritsumei.ac.jp).
Akira Maeda is with the College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan (e-mail: amaeda@media.ritsumei.ac.jp).


HOME
BROWSE CATEGORIES
ABOUT
Ask
What would you like to ask?
Continue
Ans
Home > All Categories > Entertainment & Music > Music > Other - Music > Undecided Question
Show me another »
Report Abuse

Undecided Question

What song is this? ra da da da da daaaada......?

This is repeated like a 100 times in the song. It is a very famous one. The artist is male and has a very deep voice. I just can't remember his name or the song title.

2 years ago (Tiebreaker)

Fig. 1. Example of onomatopoeia in a Q&A site

Figure 2 shows an example of representing music with onomatopoeia. More than several hundred questions asking for the name of a musical piece by using onomatopoeia have been posted on a Q&A site in Japan. This fact lead us to propose a method for retrieving music by using onomatopoeia.

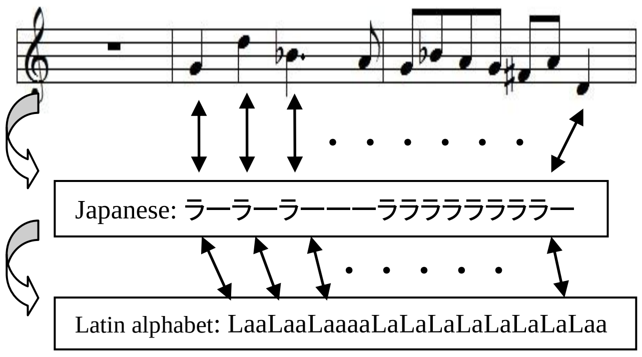

Japanese: ラーラーラーーーララララララララー
Latin alphabet: LaaLaaLaaaaLaLaLaLaLaLaLaLaa

Fig. 2. Representing music with onomatopoeia.

A. Related Work

Many kinds of systems have been proposed by many researchers in the field of music information retrieval. For example, systems that operate on "query by humming" enable intuitive search. The humming retrieval system compares a query made by humming and the features of music obtained from real audio or MIDI [1-3]. Humming retrieval systems often utilize the main melody or main vocal track from music. In other systems, there are studies to retrieve the rhythm part rather than melody [4,5]. MUSART [6] enables various kinds of retrieval with multiple features such as melody, voice, lyrics, and so on. In addition, the system creates a theme with a repeating structure of music.

There is also a system for entering the features of music, not humming. The sound retrieval system of Wake et al.[7] retrieves sound by onomatopoeia, sources (instrument) and adjective(sensitivity words). They determine those queries from experiments to ascertain the way people represent sound. Masui [8] introduces a music notation system based on onomatopoeia. The system converts melodies determined by the number of hyphens and the type of characters into MIDI. The usage of onomatopoeia in this study is very similar to our approach.

Various studies use the features of music in order to narrow down the target and to extract feature parts in music. Takeda et al.[9] proposed a method for recognizing the rhythm and tempo of a music performance on the basis of a probabilistic approach. Our retrieval system enables retrieval that targets the entire melody without requiring the user to input accurate information.

II. Outline Of The Proposed Method

In this section, we explain the outline and flow of our method. Our method consists of two parts. The first part is indexing of music data. The data is the target of the search, and we target only MIDI data. We collected MIDI data from the Web. We convert MIDI data into an index with a focus on the change in note length and pitch. The next part is the retrieval process for the user's query. We perform matching with features extracted from a query and a musical piece.

Figure 3 shows the flow of our system. First, a user enters a character string of onomatopoeia as a query. Second, the user adds the relative pitch changes of adjacent notes to the query. Third, the system calculates the similarity of each musical piece and the user's query. Last, the system presents the results ranked by the similarities to the user.

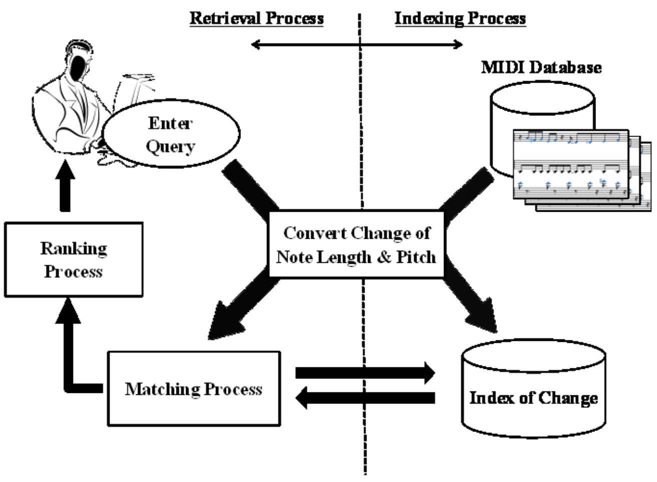


Fig. 3. Flow of our music retrieval system.

III. Indexing Process

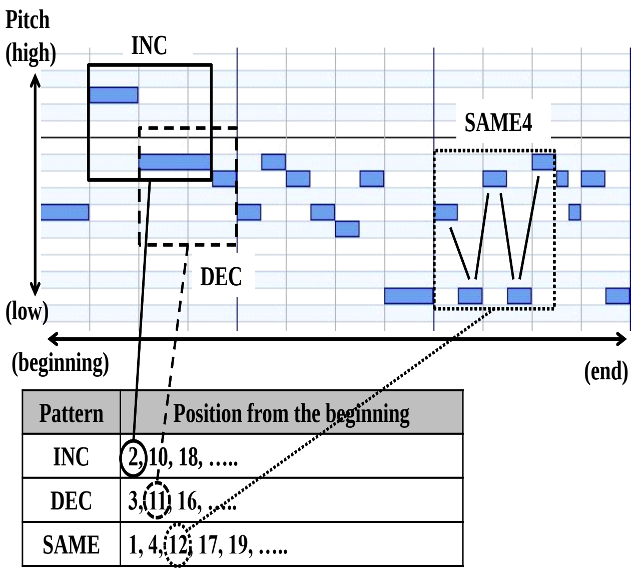
In this section, we explain the process of extracting information from MIDI data. We use mainly time series information and the pitch of the notes from a track with the main melody in the MIDI format [10]. However, it is difficult to use time series information of notes that is simply extracted from MIDI. There may be a slight difference between the sound perceived by people and the notes in the melody. For example, consider a case in which a quarter note is begun in the middle of a quarter note that is sounding. People feel that the quarter note is sounded after the eighth note in spite of the quarter note still sounding. Therefore, it is important to convert the original melody to a melody that is understood by people. In this paper, we do not consider non-melodic tracks. For instance, a drum track is not required, because most users would, in this case, enter changes in the melody characteristic rather than a regular rhythm such as a drum track.

A. *Creating Index*

The purpose of this study is to create a music retrieval system that enables searches to be performed instinctively even for users without sufficient musical knowledge. We convert music information to a simple representation in order to achieve this purpose. We create an index by focusing only on changes in the pitch and length of notes instead of the specific pitch and length of notes because we cannot obtain the pitch and length of an exact note from a string of onomatopoeia. To begin with, a person who can exactly recognize the pitch and length of the note is rare. Also, people's perception of the length of notes varies with the tempo. Our method provides flexible retrieval that reduces the effect of user mistakes.

Index of Note Length

Our method classifies the changes in length of adjacent notes into three patterns. The patterns are based on the comparison of two notes, i.e., the first note is short and the second note is long, the first note is long and the second note is short, and the two notes are the same. We denote them as "INC," "DEC," and "SAME," respectively. We create an index for these patterns. The pattern "SAME" frequently appears continuous in music. We summarize such patterns as "SAME10" if "SAME" appears continuously 10 times, for example. In addition, we register in the index only the appearance position of the beginning of "SAME10." This approach treats constant note length as a feature. Figure 4 is an example of the indexing process. These patterns have created an index on the basis of the position of the notes from the beginning of the music.



Pattern	Position from the beginning
INC	2, 10, 18, …..
DEC	3, 11, 16, …..
SAME	1, 4, 12, 17, 19, …..

Fig. 4 Indexing of note length

Index of Note Pitch

Our method classifies the change in pitch of adjacent notes into four patterns. The patterns are based on a comparison of two notes, i.e., the first note is low and the second note is high, the first note is high and the second note is low, the two notes are the same, and difficult to distinguish. We denote them as “UP,” “DOWN,” “EQUAL,” and “UNKNOWN,” respectively. Distinguishing the change in pitch is difficult if the notes used for comparison contain chords, for example, a chord comprises notes with a pitch higher and lower than the comparison note. We register the notes in the index as “UNKNOWN” in such a case. “UNKNOWN” has the potential to be one of the patterns in the other three.

IV. Retrieval Process

The retrieval process contains all of these processes after a user enters a query. We explain each process in the order of processing.

A. *Query Analysis*

Questionnaire Survey

We conducted a questionnaire with ten Japanese people. The purpose of this questionnaire was to learn how people represent melody by using onomatopoeia. We collected a hundred queries in total from users. Investigating of the queries revealed that the representation of note length varies by person, and typing errors occur in the middle of the melody and positions where notes of the same length appear continuously.

Entering Query

It is necessary to go through two steps in order to create a query. The first step is to create a character string of onomatopoeia. The second is to enter the pitch information of the notes. Figure 5 shows the interface of our system. The left side is the form of text input. The second step beings once text has been submitted. The right side of Figure 5 represents a sequence of notes that was extracted by using the melody information from the query string. One object represents one note. The user enters the pitch by dragging text squares up and down.

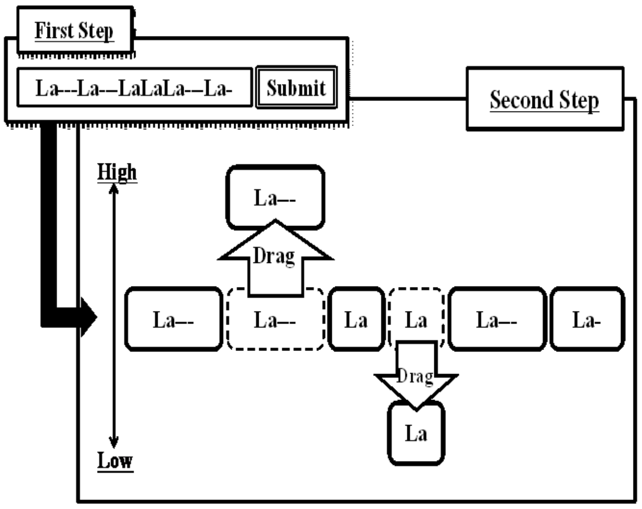


Fig. 5. Query input interface.

Converting a Query

The query entered by a user needs to be converted from the input character string into melody information. However, it is not possible to extract the exact pitch and note length from the query. Therefore our method extracts only the changes in pitch and length from the query. This process is similar to the process that we use to make the MIDI data in Section 3. The reason for focusing only on the changes is to represent that the note length varied by person in the preliminary questionnaire survey.

B. *Matching Process*

The purpose of our method in this process is to match information that has been converted from music pieces and the query. Figure 6 shows an example of how to properly search for target music pieces for the query (La---La-LaLa---La-). The complexity of the matching process becomes huge if all positions are target candidates for matching. Therefore, we conducted the following processes in order to reduce the number of time targets are compared. The position used for matching is determined by the first pattern in note length. Our system performs matching only with positions that correspond to the pattern of “DEC” in the index of all music when the first pattern of the query is “DEC.” This approach was determined from the result of questionnaire survey that showed that nine out of ten people correctly input the first pattern. In addition, we exclude a phrase from being a target for matching if a melody is interrupted or if the number of symbols is not the same. The number of patterns in the example query is four. The phrase of number 3 in Figure 6 is excluded from being a target for matching because the number of patterns is three.

Our method uses dynamic programming (DP) matching as a method of matching. We add our own rules to DP matching. In the next section, we describe the rules.

Next, we narrow targets by focusing on pitch. We calculate a “pitch score” in order to revise the “DP matching score.” Pitch information is calculated independently of the DP matching score. We compare the pitch information of a query and patterns cut out from a musical piece. The pitch score is the number of matched pitch patterns except for “UNKNOWN.”

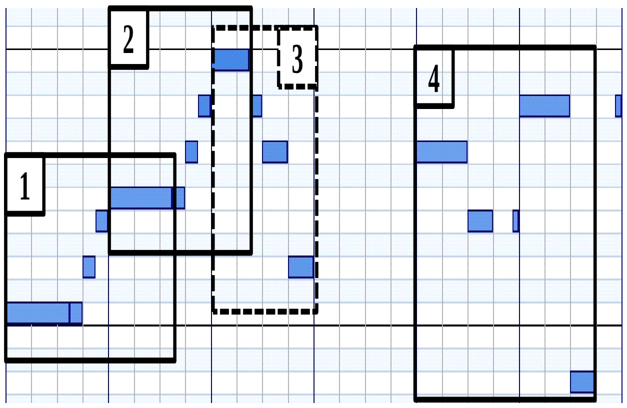


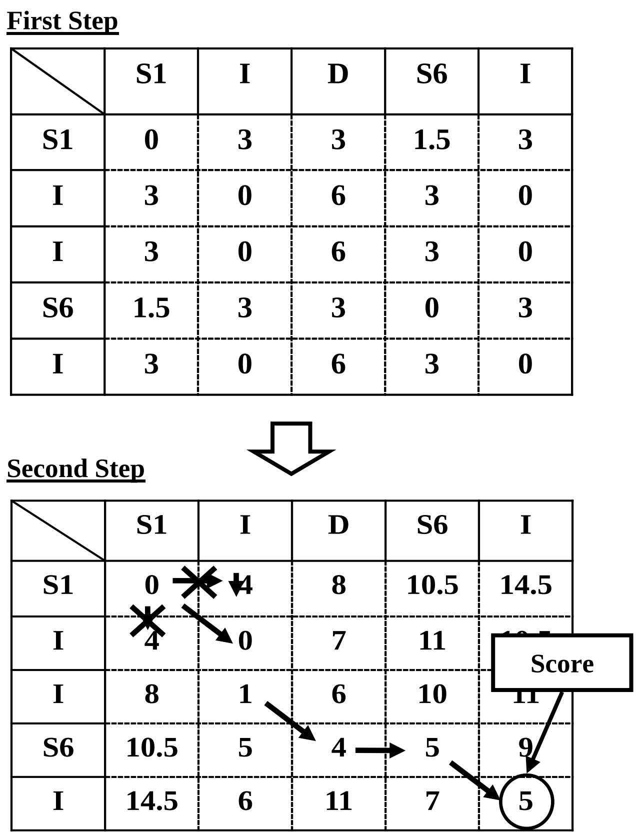
Fig. 6. Determining matching places.

Dynamic Programming Matching

DP matching is a method for measuring the similarity of patterns [11]. This method is often used for pattern matching in the field of voice analysis. Our method uses this method in order to measure the similarity of strings. The feature of this method is that it considers partial expansion and contraction, and it calculates similarity by specifying the cost to completely match characters.

The patterns that are used in DP matching are strings of symbols that represent the patterns (“INC,” “DEC,” “SAME”). We adjusted the cost of DP matching by using these symbols. It is common to set the cost so that a mismatch of characters increases the number of costs by three and inserting a character increases this number by one. We add some rules that increase cost. One of the rules is to change the cost when comparing “INC” and “SAME” and “INC” and “DEC.” We consider the probability that a user mistakes “INC” for “SAME” or “DEC.” We have to increase the cost of comparison between “INC” and “DEC” because the position of “INC” is likely to be mistaken for “SAME” than for “DEC.” A different rule is used to change the cost in the case of “SAME.” Our method abbreviates “SAME” if it appears consecutively. For example, comparing “SAME2” and “SAME5” increases the cost by three in accordance with the difference between the number of “SAME”, which is three.

Figure 7 shows an example of the processing flow of the DP matching. We explain the process of matching two strings (“S1, I, I, S6, I” and “S1, I, D, S6, I”). The first step calculates the increasing costs due to mismatching of characters. The second step determines the shortest path to consider the lowest cost for completing characters. The cost for completion increases when moving to an adjacent cell on the right and bottom. Therefore, the score in Figure 7 is five when the cost of mismatch is set to three and the cost of completion is set to one.



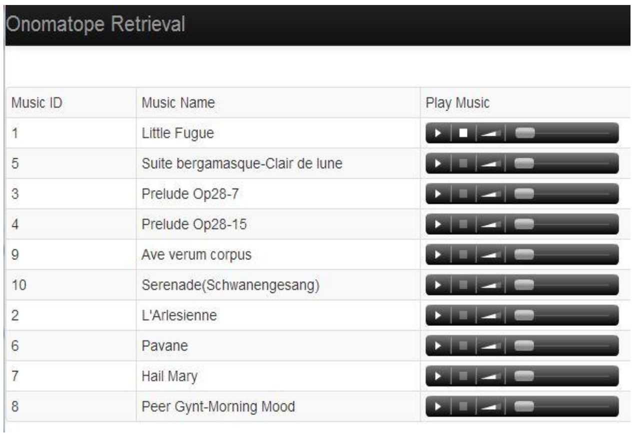
	S1	I	D	S6	I
S1	0	3	3	1.5	3
I	3	0	6	3	0
I	3	0	6	3	0
S6	1.5	3	3	0	3
I	3	0	6	3	0

	S1	I	D	S6	I
S1	0	4	8	10.5	14.5
I	4	0	7	11	10.5
I	8	1	6	10	11
S6	10.5	5	4	5	9
I	14.5	6	11	7	5

Fig. 7. Flow of DP matching

C. Ranking Process

We consider that the musical piece that has the smallest value for total cost as the correct answer. The reason is that the pattern with the lowest cost in a musical piece may have a pattern that is used by the user query. In addition, we adjust the score by using the pitch of the pattern. If a musical piece with the lowest cost has cost equal to that of another piece of music, we determine the rankings by checking next of lowest cost. This approach is applied for musical pieces in which the same phrase appears many times. Figure 8 is an example of a ranking result that our system presents to the user.


Onomatope Retrieval

Music ID	Music Name	Play Music
1	Little Fugue	
5	Suite bergamasque-Clair de lune	
3	Prelude Op28-7	
4	Prelude Op28-15	
9	Ave verum corpus	
10	Serenade(Schwanengesang)	
2	L'Arlesienne	
6	Pavane	
7	Hail Mary	
8	Peer Gynt-Morning Mood	

Fig. 8. Example ranking result

V. Evaluation

We conducted experiments to evaluate the retrieval effectiveness of our music retrieval system. Nine subjects participated in these experiments. The number of queries used in these experiments was 90 in total. We measured the average accuracy of retrieval in our method on the basis of the ranked results for 90 queries. The search target was 200 pieces of classical music.

A. Evaluation Criteria

We evaluated our method by scoring the rankings of the retrieval results for each query. The score of the method is calculated by using the mean reciprocal rank (MRR) (1).

$$\mathrm{MRR} = (1/N)\sum_{\iota=1}^{N} r_i^{-1} \tag{1}$$

, where N is the number of queries, i is the i-th query, and r is the rank of the correct answer.

In addition, we calculate the top ten score, such that the score is 1.0 if the correct piece of music is ranked at the top, the score is 0.9 if it is at the second, and the score is 0.1 if it is in 10th place. The final score of the method is the average value of all the queries.

B. Experimental Results

Table 1 shows the experimental results. The experiments were conducted for two cases, one was only using the note length, and the other was using both note length and pitch. “Length only” denotes the method of entering only the string of onomatopoeia (only the first step detailed in section 4.1.2). “Length & Pitch” denotes the method that we proposed in this paper. As shown in Table 1, using pitch information along with note length resulted in only slightly better results. Also, the number of queries put into the top 10 was 39 amongst all the queries.

Table 1. Experimental Results.

	Length Only	Length & Pitch
MRR	0.220	0.252
Top10 score	0.294	0.334
Number of top 1	12	15
Number of Top 10	35	39

C. *Discussion*

The score of our method, which was an MRR of 0.252 and top 10 score of 0.334, was not sufficient.

For comparing the evaluation, we compared our method with research on humming search that is similar in purpose. The method submitted by Ryynänen and Klapuri [3] produced an MRR of 0.885 by using Roger Jang's corpus consisting of N = 2797 eight-second queries and 48 ground-truth MIDI files. It is not possible to simply compare the result because of the difference between the query, the number of songs, and the type of music. We describe factors that cause the accuracy of our method to be low.

The first cause is errors made by users when entering an onomatopoeic string. There are different types of user mistakes. Some users did not enter a sufficient number of hyphens so that we can consider that as the change of note length. According to users, there were several opinions; e.g. judging the correspondence of notes and hyphens is difficult, and entering queries is difficult. There are some difficulties in distinguishing different note lengths by using onomatopoeia. Figure 9 shows an example of such an error. If a user interprets a quarter note as "La-," the eighth note in the example is most likely described as "La." In that case, the 16th note cannot be described because there is no character that represents a note length shorter than "La."

The second cause is the reduction in the number of features. Narrowing down the number of candidates for a correct musical piece is not enough when supporting vague user input. This problem often appears when the user's query is short. Half of our queries are shorter than five seconds, whereas most of the queries that Ryynänen et al. [4] used in their experiment were about eight seconds. The length of the query is different subjects by subjects. The score of MRR for the three subjects was lower than 0.1. The number of short queries of these subjects was more than seven in each subject. In such cases, the difference of scores by DP matching is very small. There are two approaches to solving this problem. The first is to increase the number of features, using features other than the change in note length and pitch. The second is to recommend the user to enter a long query.

However, the experiments of these two cases do not show a big difference in scores. This difference in score is not a problem because the case for "Length & Pitch" in Table 1 is based on the case of "Length Only." The change in these cases is that they rank musical pieces by the pitch score in addition to ranking by "Length Only" if there are musical pieces with the same score. However, the results of the experiments look different when comparing queries one by one. The ranking of each query is up or down when comparing the experiments in the two cases. There were users going to enter a query even when the change in pitch was ambiguous. We think that this problem can be solved by implementing an interface that can check the change of the sound on input. Also, we should recommend the user to enter only positions that the user is sure about the correct answer. In other cases, entering "UNKNOWN" is not used for score calculation.

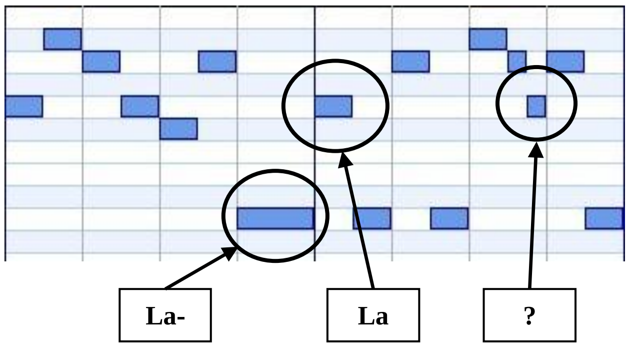


Fig. 9. Errors in entering onomatopoeia.

VI. Conclusion

In this paper we proposed a music retrieval system that uses onomatopoeia query, enabling easy and intuitive retrieval of music even for users without musical knowledge. This system enables music to be searched for even if the user does not have information on the music and had just heard it. The advantage of our method is that it retrieves music in consideration of mistakes made by the user by ambiguous input. However, the retrieval accuracy of our method is not enough because it focuses only on changes of pitch and length, and it reduces the number of features needed to narrow down the music. It is necessary to conduct experiments by using large music databases such as [12].

As future work, we plan to use the feature that is included in the character of onomatopoeias in the retrieval process. We think that people use different characters for onomatopoeias for different sounds. The feature is not useful for accurately judging a melody. However, it is possible to grasp the tendency of the melody. For example, we think it is possible to determine the difference in pitch or instruments.

There is room for improvement also on how to narrow down the melody of the correct answer. Our approach is to narrow down only the features of the rhythm. Therefore, it is necessary to narrow down the candidates by incorporating the pitch information into the matching process.

In addition, we are planning to improve the interface for entering queries. We obtained opinions that entering onomatopoeias by typing characters is difficult and cumbersome. Also, entering a query by typing is the primary factor that generates errors. It is desirable to implement an interface in which a user can enter a query more easily and intuitively.

References

[1] C. Bandera, A. M. Barbancho, L. J. Tard´on, S. Sammartino and I. Barbancho: "Humming Method for Content-Based Music Information Retrieval," *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 49-54, 2011.

[2] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamoto and K. Kushima: "A Practical Query-By-Humming System for a Large Music Database," *Proceedings of the eighth ACM international conference on Multimedia* Pages 333-342, 2000.

[3] M. Ryynänen and A. Klapuri: "Query by humming of midi and audio using locality sensitive hashing," In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2008, pp.2249-2252.

[4] G. Tzanetakis, A. Kapur, and M. Benning: "Query-by-Beat-Boxing: Music Retrieval For The DJ," *In Proceedings of the 5th International Conference on Music Information Retrieval, 2004.*

[5] T. Nakano, J. Ogata, M. Goto, Y. Hiragi: "A Drum Pattern Retrieval Method by Voice Percussion," *In Proceedings of the 5th International Conference on Music Information Retrieval, 2004.*

[6] W. P. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. Bartsch,D. Bykowski, D. Mazzoni, C. Meek, M. Mellody and W. Rand: "MUSART: Music Retrieval Via Aural Queries," *3rd International Conference on Music Information Retrieval(2001).*

[7] S. Wake and T. Asahi: "Sound Retrieval with Intuitive Verbal Expressions," *ICAD'98 Proceedings of the 1998 international conference on Auditory Display, pages30-30, 1998*

[8] T. Masui: "Music composition by onomatopoeia,*" In proceeding of: Entertainment Computing: Technologies and Applications, IFIP First International Workshop on Entertainment Computing (IWEC 2002), 2002.*

[9] H. Takeda, T. Nishimoto and S. Sagayama: "Rhythm and Tempo Recognition of Music Performance from a Probabilistic Approach," *In Proceedings of the 5th International Conference on Music Information Retrieval, 2004.*

[10] C. McKay: "Automatic genre classification of MIDI recordings," *Master's thesis. McGill University, Canada, 2004.*

[11] H. Sakoe and S. Chiba: "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. ASSP-26, NO.* ***1,*** *FEBRUARY 1978.*

[12] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka: "RWC Music Database: Popular, Classical, and Jazz Music Databases," *In Proceedings of the 5th International Conference on Music Information Retrieval,* pp.287-288, 2002.