MusicMixer: Automatic DJ System Considering Beat and Latent Topic Similarity

Tatsunori Hirai[1(✉)], Hironori Doi[2], and Shigeo Morishima[3,4]

[1] Waseda University, Tokyo, Japan
tatsunori_hirai@asagi.waseda.jp
[2] Dwango, Tokyo, Japan
[3] Waseda Research Institute for Science and Engineering, Tokyo, Japan
[4] JST CREST, Tokyo, Japan

Abstract. This paper presents *MusicMixer*, an automatic DJ system that mixes songs in a seamless manner. MusicMixer mixes songs based on audio similarity calculated via beat analysis and latent topic analysis of the chromatic signal in the audio. The topic represents latent semantics about how chromatic sounds are generated. Given a list of songs, a DJ selects a song with beat and sounds similar to a specific point of the currently playing song to seamlessly transition between songs. By calculating the similarity of all existing pairs of songs, the proposed system can retrieve the best mixing point from innumerable possibilities. Although it is comparatively easy to calculate beat similarity from audio signals, it has been difficult to consider the semantics of songs as a human DJ considers. To consider such semantics, we propose a method to represent audio signals to construct topic models that acquire latent semantics of audio. The results of a subjective experiment demonstrate the effectiveness of the proposed latent semantic analysis method.

Keywords: DJ system · Song mixing · Latent topic analysis · Beat similarity · Machine learning

1 Introduction

Many people enjoy listening to music. The digitalization of music content has made it possible for many people to carry their favorite songs on a digital music player. Opportunities to play such songs are frequent, e.g., at a house party or while driving a car. At some parties, an exclusive DJ performs for the attendants. DJs never stop playing the music until the party ends. They control the atmosphere of the event by seamlessly mixing songs[1]. However, it is not always realistic to personally hire a DJ. Thus, we present *MusicMixer*, an automatic DJ system that can mix songs for a user.

One of the most important things in a DJ's performance is to mix songs as naturally as possible. Given a list of songs, a DJ selects a song with beats and

[1] The word "mix" here refers to the gradual transiton of one song to another.

Q. Tian et al. (Eds.): MMM 2016, Part I, LNCS 9516, pp. 698–709, 2016.
DOI: 10.1007/978-3-319-27671-7_59

sounds that are similar to a specific point in the currently playing song such that the song transition is seamless. Consequently, the songs will be mixed as a consecutive song. The beats are particularly important and should be carefully considered. Maintaining stable beats during song transition is the key to realizing a seamless mix. The time to select the next song is limited and the songs are numerous; therefore, many DJs intuitively select a song to connect. However, this might not be the best song. The innumerable possibilities of mixing songs make performing difficult for the DJ.

Computers are good at searching for the best pairs of beats from innumerable possibilities. It is possible to solve this problem using a signal processing technique to extract beats and rely on a computer to retrieve a similar beat for effective mixing. However, computers handle audio signals numerically without considering the underlying song semantics; thus, the resulting mix will be mechanical if the system only considers beat similarity. The latent semantics of songs must be considered in addition to the beats. The DJ attempts to switch to a new song when the two songs sound similar. To consider the latent semantics, we propose a method to analyze the latent topics of a song from the polyphonic audio signal. These topics represent latent semantics about how chromatic sounds are generated. In addition to beat similarity, the proposed system considers the similarity of latent topics of songs. In particular, by employing a machine learning method called latent Dirichlet allocation (LDA) [1], the proposed system infers latent topics that generate chromatic audio signals. This process corresponds to consideration of how sound is generated from latent topics in a given song. By inferring similarity among song topics, higher level song information can be considered.

MusicMixer takes advantage of computational machine power to retrieve a good mix of songs. To make mixing more seamless as the DJ mix, the system focuses on the similarity of latent topics in addition to beat similarity and realizes natural song mixing (Fig. 1).

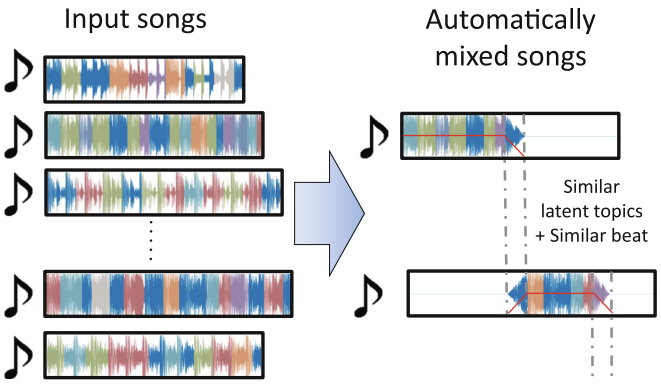

Fig. 1. Conceptual image of mixing songs with similar latent topics and beats using MusicMixer

2 Related Work

2.1 Music Mixing and Playlist Generation

Ishizaki *et al.* proposed a DJ system that adjusts the tempo of songs [2]. They defined a measurement function for user discomfort relative to tempo adjustment based on a subjective experiment. However, that system only considers tempo and beat. In addition, their system does not retrieve a mixing point but forcibly changes the tempo of songs.

Several studies have focused on generating a music playlist [3–6]. AutoDJ generates a playlist based on one or more seed songs using Gaussian process regression [3]. The AutoDJ project team has also proposed a method to infer the similarity between music objects and have applied this to playlist generation [6]. However, these approaches focused on playlist generation, and the importance of mixing (connecting) songs was not considered.

Goto *et al.* proposed *Musicream* [7], which provides a novel music listening experience, including sticking, sorting, and recalling musical pieces. It also provides a playlist generation function; however, mixing is not considered.

There is another approach to mixing songs, referred to as mashup. Mashup creates a single song from multiple songs. AutoMashUpper [8] generates a mashup according to a mashability measure. Tokui proposed an interactive music mashup system called Massh [9]. Mashups are a DJ track composition style; however, not all DJs can perform mashups live without using a pre-recorded mashup song. The mainstay of a DJ performance is mixing.

However, there has been little research on DJ mixing comparing to the research on playlist generation. We believe that a mixing method combined with playlist generation methods could be a powerful tool. In this paper, we propose a DJ system for mixing songs that considers beats and the higher level information of a song. During a DJ performance, the higher level music information that should be considered is the semantics of songs.

2.2 Topic Modeling

In natural language processing research, there is a method called topic modeling which estimates the topic of a sentence from words that appear in the sentence. Those words depend on the topic of the sentence; thus, the topic can be estimated by observing the actual sentence. If the topics are the same for two sentences, the sentences will be similar at a higher semantic level. Sasaki *et al.* proposed a system to analyze latent topics of music via topic modeling of lyrics [10]. They proposed an interface to retrieve a song based on the latent topics of lyrics.

Topic modeling can be applied to actual features. In this case, feature vectors should be quantized (e.g., a bag-of-features) [11]. Nakano *et al.* applied topic modeling to singing voice timbre [12]. They defined a similarity measure based on the KL-divergence of latent topics and showed that singers with similar singing voices have similar latent topics. However, it is difficult to understand the meaning of each topic explicitly using feature vectors rather than words.

Hu *et al.* used the note names of a song as words to estimate the musical key of a song using topic modeling [13]. This shows that topic analysis using note names is effective for inferring the latent semantics of a song. Hu *et al.* also proposed an extended method to estimate musical keys from an audio signal using a chroma vector (i.e., audio features based on a histogram of a 12 chromatic scale) rather than note names [14]. This approach shows that topic modeling using a chroma vector is useful for inferring the latent topic of a song.

3 System Overview

MusicMixer requires preprocesses to analyze song beats and latent topics. The beat analysis is performed using an audio signal processing approach. Figure 2 shows the system flow.

First, a low-pass filter (LPF) is applied to the input song collection to extract low-frequency signals. In the low-frequency signal, beat information such as bass and snare drums and bass sounds is prominent. Thus, beat information can be acquired by detecting the peaks of the envelope in the low-frequency signal.

The latent topic analysis is realized using LDA [1]. First, the system constructs a topic model using a music database that includes various music genres. The latent topics for a new input song can be estimated using the constructed

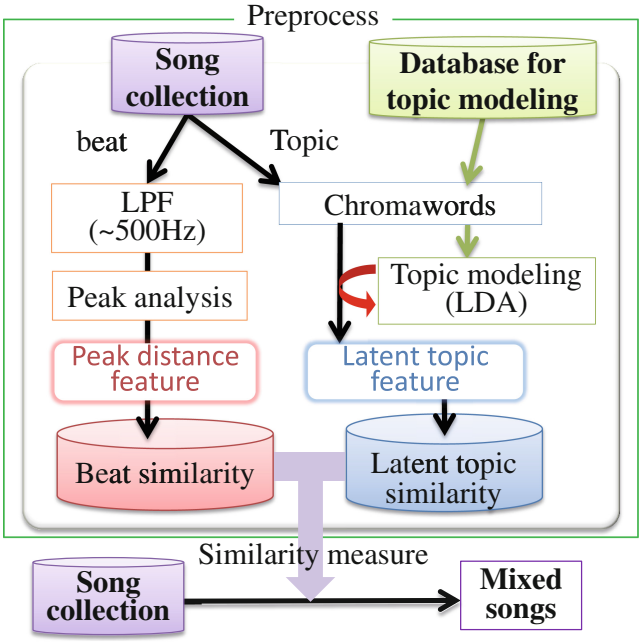

Fig. 2. System flow

model. Our goal is to find a good mixing point rather than analyze the topic of a whole song; thus, we analyze the topics of segmented song portions.

Finally, the system retrieves the most similar song fragments based on the combination of beat similarity and latent topic similarity. Once a similar pair of song fragments is retrieved, the system mixes songs at the fragment by cross-fading (i.e., fading in and out). Thus, the songs are mixed naturally. To mix more songs for endless playback, the similarity-based retrieval is applied to the mixed song.

4 Beat Similarity

In particular, the sound of the bass drum plays a significant role (e.g., the rhythm pattern called four-on-the-floor is composed of bass drum sounds). In addition to the bass drum, the snare drum and other bass sounds are important to express detailed rhythm. Note that we assume that all the other sounds do not affect to the beat. To ignore other audio signals, we apply an LPF, which passes signals with frequency below 500 Hz. The LPF passes the attack sounds of a general snare drum. By analyzing the peaks of the envelope of a low-frequency signal, dominant sound events in the low-frequency spectrum, such as the attack of a bass drum, can be detected. The distances between peaks correspond to the length of the beat (Fig. 3).

The beat similarity is calculated by comparing the distances between N peaks of the envelope. Here, N is the number of peaks to consider. The peak distance feature D_{peak} is an N dimensional vector. The beat similarity S_{beat} between fragment i and fragment j is calculated as follows:

$$S_{beat}(i,j) = \frac{1}{\sum_{k=1}^{N-1} ||D^i_{peak}(k) - D^j_{peak}(k)|| + 1}. \quad (1)$$

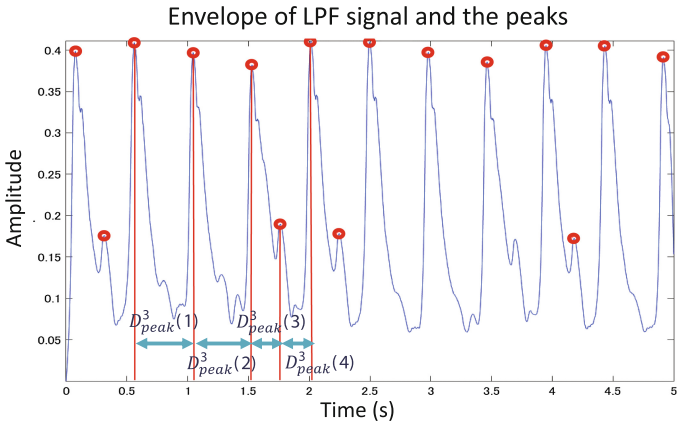

Fig. 3. Extracting peak distance features from the envelope of low frequency audio signal

Here, larger N values realize better matching relative to beats. However, the number of candidate songs to be mixed will be excessively reduced if the N value is too large. At present, this parameter is user-defined.

5 Latent Topic Similarity

This section describes the method to analyze a latent topic of a song using topic modeling. In particular, we propose a topic modeling method that considers the latent topic of a song by expressing the audio signal symbolically.

5.1 Topic Modeling

The topic model is constructed by extracting the features of songs and applying LDA [1] to the features. We extract the chroma vector from the audio signal and represent the feature symbolically, which we refer to as "ChromaWords."

Extraction of ChromaWords. Latent topics typically include semantics. However, topic analysis using audio feature values makes it difficult to understand the meaning of topics explicitly. It is difficult to determine meaning from a high dimensional feature value; therefore, previous topic modeling methods could not describe the meaning of each topic clearly. To avoid losing topic meaning, we express the audio signal symbolically. There are symbols in music that are represented in a musical score, e.g., note names. Since a letter is assigned to each note, we can use the letters to construct a word for topic modeling.

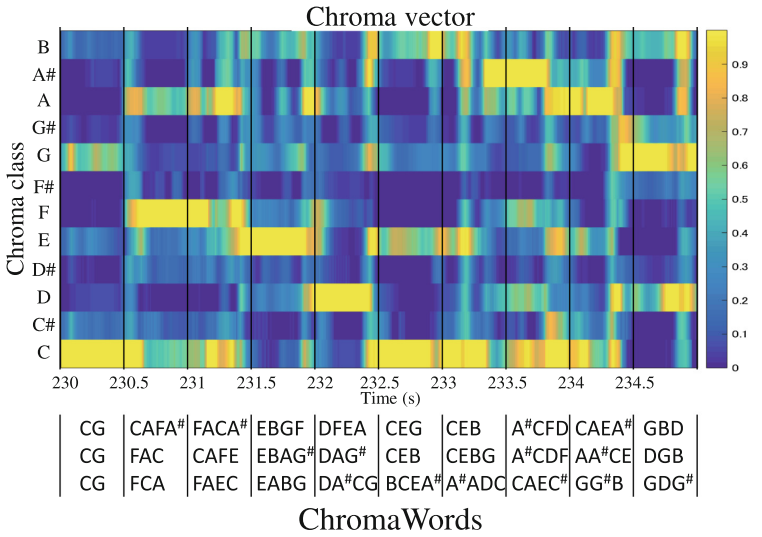

Fig. 4. Extraction of ChromaWords from chroma vector

Here, we employ an audio feature referred to as a chroma vector, which is a histogram of 12 notes. Each bin of the chroma vector represents a musical note. By sorting the chroma vector by dominant notes, a word can be generated (e.g., [CADE], [BAD#]) which we refer to as ChromaWords. Typically, a chroma vector includes noise caused primarily by inharmonic sounds. To avoid the effects of the noise, we use the top 70 % power of notes. Here, we set the maximum length of the word to four letters. Thus, we can represent polyphonic audio signals symbolically with natural language processing. Figure 4 shows an example of ChromaWords (bottom) acquired from the chroma vector of an actual song (top). Note that "#" is not counted as a single letter. Because of space limitations, we only display three ChromaWords per 0.5 s. These three ChromaWords are sampled at equal interval (0.5 s). The leftmost letter is the most dominant component, and less dominant components are to the right.

ChromaWords are acquired per audio frame. Here, the audio sampling rate is 16000 Hz monaural, and the frame length is 200 ms, shifting every 10 ms. One hundred words can be acquired from a 1-s audio signal.

Latent Dirichlet Allocation. By acquiring ChromaWords from a song, topic modeling can be applied similar to methods in natural language processing. MusicMixer employs LDA [1] for training of latent topic analysis. The number of topics is set to 100 in order to express semantics more complex than those of basic western tonality. The vocabulary of ChromaWords is 13345 ($= {}_{12}P_4 + {}_{12}P_3 + {}_{12}P_2 + {}_{12}P_1 + 1$), including perfect silence.

Training is required prior to latent topic analysis. We use 100 songs from the RWC music database [15], which comprises of songs of various genres. The parameters and algorithm for LDA is the same as the topic modeling method employed for the latent topic analysis of lyrics [10].

Table 1 shows the top-five representative ChromaWords for each topic learned from the RWC music database (10 topics out of 100, sorted by probability). The leftmost letter in a ChromaWord indicates the dominant note in the sound. Because many initial letters in ChromaWords for the same topic are the same, the topic model constructed by LDA reflects the semantics of chromatic notes, which was difficult for previous methods to explicitly express.

Using the constructed model, the latent topics for a new input song can be estimated by calculating a predictive distribution. The latent topics for the new input song are represented as a mixing ratio of all 100 topics. Figure 5 shows

Table 1. Top-five ChromaWords allocated to each topic.

Topic22	Topic90	Topic7	Topic98	Topic78	Topic52	Topic9	Topic79	Topic43	Topic80
CBC#A	Silent	AG	CFG	AA#BC	ED	AFB	Silent	AEA#	DAA#C
CC#BA	GDCA	AA#G	CFGD	ABA#C	E	AA#BF	GA	AEA#B	DACA#
CC#AB	GCDA	AGA#	CGFA	BAA#C	F	Silent	CF	Silent	ADA#C
CBAA#	GDAC	ADGA#	CGFD	AA#CB	DE	AA#GB	CFAG	AEA#G	DACC#
CABA#	DA	ADA#G	CGF	ABCA#	EF	AA#FB	FC	AEA#C	AA#BC

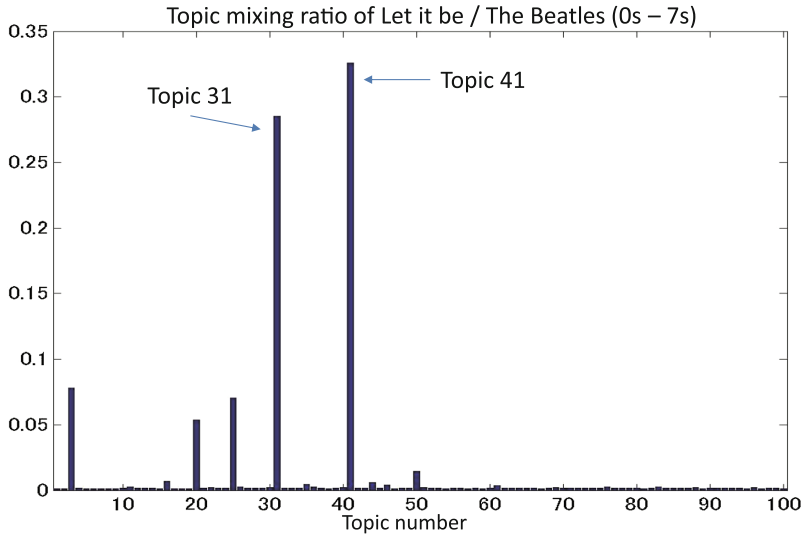

Fig. 5. Results of latent topic analysis applied to a 7-s fragment of the song "Let it Be"

an example of topic analysis for a fragment of the song "Let it Be" by The Beatles. In this result, topic 41 includes ChromaWords "DB," "GB," and "AC" as dominant words, and the dominant letters in the ChromaWords of topic 31 are "F," "C," and "A." In fact, the chord progression of this part of the song is "C, G, Am, F," which shows that topics mostly reflect the notes consisted in these chords. This indicates the relevance between chords and ChromaWords. Note that chords or harmony effect the ChromaWords, but the topics themselves do not directly represent chords or harmony.

Here, our goal is to find a good mixing point rather than analyze the topics of an entire song. Therefore, MusicMixer analyzes latent topics every 5 s to acquire the temporal transition of the topic ratio.

5.2 Calculation of Latent Topic Similarity

The mixing ratio of latent topics for each 5-s song fragment is acquired by the above-mentioned method. The mixing ratio is extracted as 100-dimensional feature vectors, and we use the mixing ratio as the latent topic feature f.

The latent topic similarity S_{topic} between fragment i and fragment j is calculated in the same form as the beat similarity:

$$S_{topic}(i,j) = \frac{1}{\sum_{k=1}^{K} ||f_i(k) - f_j(k)|| + 1}, \tag{2}$$

where K is the number of topics (100).

5.3 Evaluation

We performed a subjective evaluation experiment to evaluate the effectiveness of latent topic analysis using ChromaWords. We compared the proposed method to a latent topic analysis method using MFCC feature values and chroma vector feature values. The topic modeling method for the compared methods is based on the method proposed by Nakano *et al.* [12], which uses k-means clustering to describe feature values in a bag-of-features expression. Note that we do not use similarity calculated from raw feature values; thus, we can focus on the effects of ChromaWords.

Fifty pop, rock, and dance songs were used in the experiment, and 2192 segments were generated by cutting the songs into 5-s fragments. We calculated the latent topic similarity of all pairs between the 2192 fragments.

The subjects were asked to listen to two pairs of song and indicate which pair was more similar. A pair was generated based on the latent topic similarity of each method. We selected three pairs of songs per method. The three pairs were selected from the top-30 latent topic similarity. Note that song repetition was avoided in this experiment. To avoid the effects of beat, we did not mix songs but played each song separately.

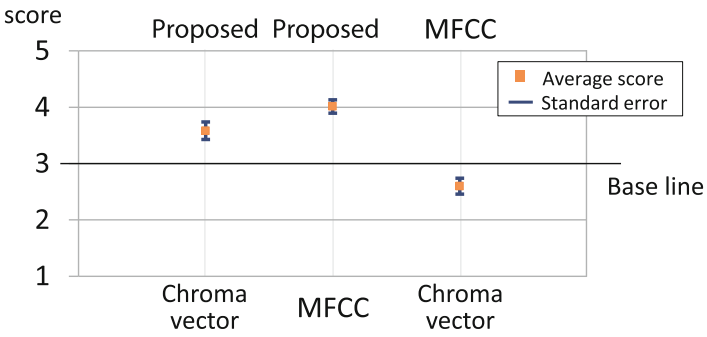

Fig. 6. Result of subjective evaluation experiment

Eight students (ages 22 to 24) with no DJ experience participated in the experiment. The subjects listened to a song pair generated by method A first, and then a pair generated by method B. They then rated the pairs from 1 (Pair A is more similar than B) to 5 (Pair B is more similar than A). A score of 3 indicates that "both pairs are equal in terms of similarity." The pairs were presented randomly.

Figure 6 shows the results of the experiment. The score is the average of all eight subjects and all nine compared pairs. Comparing the proposed topic modeling method with the topic modeling method using raw chroma vector feature, the score was 3.58, which indicates that the proposed method expresses similarity better. Comparing the proposed method with the topic modeling with

MFCC, the score was 4.01, which also indicates that the proposed method performs better. The rightmost plot shows a comparison of MFCC and chroma vector methods, which are not related to the proposed method. As can be seen, the topic model using the chroma vector outperforms the MFCC method. These results indicate that the proposed method using ChromaWords outperformed the other methods in terms of music fragment similarity.

6 Mixing Songs

MusicMixer mixes songs based on the similarity measurements described above. The combined similarity S between fragments i and j can be calculated as follows:

$$S(i,j) = w \times S_{beat}(i,j) + (1-w) \times S_{topic}(i,j), \tag{3}$$

where w denotes the weight parameter used to change the balance of the beat and latent topic similarity ($w = 0.5$ in the current implementation).

The length of each song fragment depends on the number of peaks in the beat similarity calculation. Although the length of each fragment differs, beat similarity ensures that fragments with similar length are selected as similar beat fragments. In addition, the fragment lengths are not 5 s (the length for topic analysis). Therefore, we assume that the fragment in a 5-s fragment is similar relative to the latent topic feature. Thus, we use the same latent topic feature even though the length of the fragment is not 5 s.

There is a function to specify the scope of when mixing can occur. For example, we do not want to switch to a new song during the beginning of the previous song or start a song at the end. In the current implementation, a song will not change until the latter half, and a song will start no later than the first half.

7 Discussion

7.1 Limitations

MusicMixer considers both beat and latent semantics. However, latent semantics are limited to the chromatic audio signal. Therefore, other types of high level information such as variation of instrument or dynamics within a song cannot be considered in the current implementation. In future, we will explore the possibility of semantic topic analysis using the symbolic representation acquired from the audio signals.

MusicMixer does not consider lyrics or their semantics. Therefore, a summer song may be selected after a winter song, which is undesirable. It is possible that a user compensates for such flaws by introducing user interface.

7.2 Applications of MusicMixer

MusicMixer's applications are not limited to an automatic DJ tool. There is a style of DJ performance referred to as "back-to-back" which is collaborative play among multiple DJs. In a back-to-back session, a partner DJ selects the next song while one DJ's song is playing. Thus, the partner DJ's play may be unpredictable. Although the back-to-back style cannot be performed alone, a DJ system such as MusicMixer could act as a partner DJ for a back-to-back performance. This is similar to playing a video game against the computer, which can improve the player's technique. Furthermore, collaboration with a computer might produce new or unexpected groove.

It is also possible for inexperienced people to practice DJ performance using MusicMixer. For example, mixing songs is the difficult part of a DJ's performance, but song selection might be easier for inexperienced people. In this case, the connection of songs could be performed by the system, and the user can focus on song selection. Conversely, the user can focus on song mixing without worrying about song selection by allowing system to select songs. DJ performance requires significant skill that can be only acquired from practical experience.

7.3 Conclusion

We have presented MusicMixer, an automatic DJ system. We have proposed a method to mix songs naturally by considering both beat and latent topic similarity. Our main contribution is the application of topic modeling using ChromaWords, which are an audio signal-based symbolic representation. Previous topic modeling methods have analyzed the latent topics of audio or images using features represented as a bag-of-features, so the meaning of topic was not clear. We have achieved topic modeling with understandable topic meanings using ChromaWords. Furthermore, the results of a subjective evaluation indicate that our topic modeling method outperforms other methods in terms of music similarity. Topic modeling is primarily used to analyze latent semantics in observed data. The proposed method makes it possible to employ the latent semantics of chromatic sounds. However, the semantics of chromatic sounds do not cover all the semantics of a song. Thus, in future, we plan to consider other semantics such as timbre.

This study has focused on song mixing without changing the original songs. In a future implementation, the proposed system will perform modulation of songs to allow free connection of any type of song pairs. For example, by using a song morphing method [16], it may be possible to embed such a function. In addition, we plan to consider the structure of songs. In this manner, a new song will not be played until the verse of the current song ends. We will explore the possibility of human and computer collaborative DJ performance.

Acknowledgments. This work was supported by OngaCREST, CREST, JST and JSPS Grant-in-Aid for JSPS Fellows. This work was inspired by Tonkatsu DJ Agetaro.

References

1. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Ishizaki, H., Hoashi, K., Takishima, Y.: Full-automatic DJ mixing system with optimal tempo adjustment based on measurement function of user discomfort. In: Proceedings of ISMIR, pp. 135–140 (2009)
3. Platt, J. Burges, C., Swenson, S., Weare, C., Zheng, A.: Learning a gaussian process prior for automatically generating music playlists. In: Proceedings of NIPS, pp. 1425–1432 (2001)
4. Aucouturier, J.J., Pachet, F.: Scaling up music playlist generation. In: Proceedings of ICME, pp. 105–108 (2002)
5. Pampalk, E., Pohle, T., Widmer, G.: Dynamic playlist generation based on skipping behavior. In: Proceedings of ISMIR, pp. 634–637 (2005)
6. Ragno, R., Burges, C., Herley, C.: Inferring similarity between music objects with application to playlist generation. In: Proceedings of MIR, pp. 73–80 (2005)
7. Goto, M., Goto, T.: Musicream: integrated music-listening interface for active, flexible, and unexpected encounters with musical pieces. Inf. Media Technol. **5**(1), 139–152 (2010)
8. Davies, M., Hamel, P., Yoshii, K., Goto, M.: AutoMashUpper: automatic creation of multi-song music mashups. Trans. Audio Speech Lang. Process. **22**(12), 1726–1737 (2014)
9. Tokui, N.: Massh!: a web-based collective music mashup system. In: Proceedings of DIMEA, pp. 526–527 (2009)
10. Sasaki, S., Yoshii, K., Nakano, T., Goto, M., Morishima, S.: LyricsRadar: a lyrics retrieval system based on latent topics of lyrics. In: Proceedings of ISMIR, pp. 585–590 (2014)
11. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: Proceedings of ICCV, pp. 1470–1477 (2003)
12. Nakano, T., Yoshii, K., Goto, M.: Vocal timbre analysis using latent Dirichlet allocation and cross-gender vocal timbre similarity. In: Proceedings of ICASSP, pp. 5202–5206 (2014)
13. Hu, D., Saul, L.: A probabilistic topic model for unsupervised learning of musical key-profiles. In: Proceedings of ISMIR, pp. 441–446 (2009)
14. Hu, D., Saul, L.: A probabilistic topic model for music analysis. In: Proceedings of NIPS (2009)
15. Goto, M.: Development of the RWC music database. In: Proceedings of ICA, pp. 553–556 (2004)
16. Hirai, T., Sasaki, S., Morishima, S.: MusicMean: fusion-based music generation. In: Proceedings of SMC, pp. 323–327 (2015)