LYRICLISTPLAYER: A CONSECUTIVE-QUERY-BY-PLAYBACK INTERFACE FOR RETRIEVING SIMILAR WORD SEQUENCES FROM DIFFERENT SONG LYRICS

Tomoyasu Nakano **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{t.nakano, m.goto}[at]aist.go.jp

ABSTRACT

This paper presents *LyricListPlayer*, a music playback interface for an intersong navigation and browsing that enables a set of musical pieces to be played back by music zapping based on lyrics words. In other words, this paper proposes a novel concept we call *consecutive-query-by-playback*, which is for retrieving similar word sequences during music playback by using lyrics words as candidate queries. Lyrics can be used to retrieve musical pieces from the perspectives of the meaning and the visual scene of the song. A user of LyricListPlayer can see time-synchronized lyrics while listening, can see word sequences of other songs similar to the sequence currently being sung, and can jump to and listen to one of the similar sequences. Although there are some systems for music playback and retrieval that use lyrics text or time-synchronized lyrics and there is an interface generating lyrics animation by using kinetic typography, LyricListPlayer provides a new style of music playback with lyrics navigation based on the local similarity of lyrics.

1. INTRODUCTION

Since a song's lyrics can be used to convey emotions/passions/feelings/thoughts and to facilitate imagine visual scenes, they are an important element helping listeners have emotional involvement to the song. In fact, some listeners are aware of lyrics while listening to music and use them as a criterion for selecting musical pieces [1].[1] Lyrics are text-based information and can be used as a retrieval query by music professionals and casual listeners. Indeed, there are many works focusing on how to retrieve/browse music by using lyrics [2–7].

Previous works investigating the use of lyrics in music information retrieval have focused on the following three approaches

- 1) keyword-based retrieval – Retrieving lyrics by using text-based keywords of music search web sites[2] . Retrieves based on a full-text search using lyrics text or metadata such as song titles are available, as are retrieves based on a released year or a decade, music genre, scene (supportive, love, spring, or summer), ranking, and comments from listeners. SyncPlayer [3], a query-by-lyrics retrieval system, can navigate from a list of retrieved results to the corresponding matching positions within the audio recordings.
- 2) content-based retrieval (song-level lyrics similarity/classification) – Retrieving/browsing lyrics by favorite lyrics via query-by-example systems. Lyrics can be used to retrieve songs by visualizing music archives [4, 6, 7] and recommended songs [8]. Automatic topic detection [2, 7, 9, 10] and semantic analysis [11] of song lyrics have also been proposed. Several approaches analyzed the text of lyrics by using natural language processing to classify lyrics according to emotions, moods, and genres [12–15].
- 3) hyperlinking lyrics [5] – Creating a hyperlink from a word sequence in the lyrics of a song to the same sequence in the lyrics of another song and using the hyperlink for navigating/discovering lyrics.

We propose a music playback interface, *LyricListPlayer*, that is based on an extension of a hyperlinking lyrics system [5]. The paper describing that the previous system focused on creating keyword-based hyperlinks without interaction and just mentioned using the hyperlinking structure as a basis for imaging applications. In contrast, we focused on creating similarity-based hyperlinks with interaction to

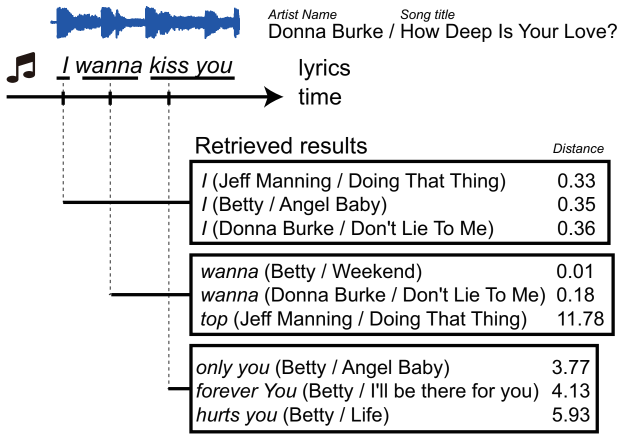

Figure 1. Consecutive-query-by-playback: LyricListPlayer uses lyrics (word sequences) currently being sung as candidate queries, and similar word sequences from different song lyrics are immediately updated during music playback.

[1] In their questionnaire investigation, 66 of 86 subjects said they are usually conscious of lyrics while listening to music, and 42 of 86 subjects said they often choose songs based on lyrics [1].

[2] *e.g.*, MusiXmatch https://www.musixmatch.com/

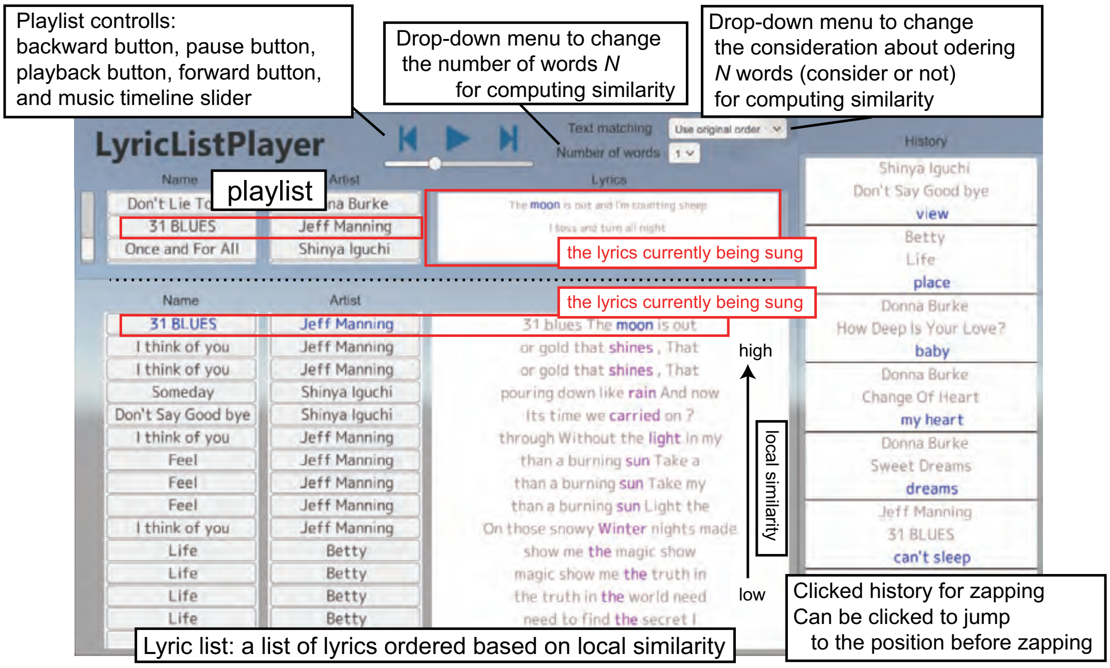

Figure 2. An example LyricListPlayer screen. When the query (a word currently being sung) is "moon", the interface can retrieve "shines," "rain," "light," and "sun."

increase a potential of the system. LyricListPlayer computes word-sequence-level similarity to cover linking not only identical word sequences but also similar sequences. It also increases the flexibility of retrieval by letting users change the length (number of words) and order of the sequence used for computing similarity.

Thus, LyricListPlayer has a potential to provide a new immersive style of music playback on the *Music Web* where songs are hyperlinked to each other on the basis of their lyrics [5]. Lyrics (word sequences) currently being sung can be issued as candidate queries automatically, and retrieved results are immediately updated during music playback. We call this novel concept of music information retrieval, consecutive-query-by-playback (Fig. 1).

The interface displays time-synchronized lyrics and uses lyrics (words) of a song currently being played back as a query. LyricListPlayer can also retrieve local similar lyrics and they can be played back to check sung style (vocal timbre and melody) and/or sung context (story of the lyrics). Similar lyrics, which like the currently sung lyrics are changed from moment to moment, are also displayed and they can be clicked to listen to them immediately. To compute lyrics similarity, latent meanings (topics) behind the words are estimated. The interface can retrieve words that are in some way similar to a query word. When the query is "angel", for example, the interface can retrieve "snuggle" and "love."

2. LYRICLISTPLAYER: AN INTERFACE FOR QUERY CANDIDATES GENERATION BY MUSIC PLAYBACK

LyricListPlayer is a music playback interface for a set of songs, and similar word sequences from the song currently being played back are displayed. Interaction and hyperlinked relationships between songs can provide a new perspective as a combination of a passive music retrieving interface and an active music listening interface [16], a combination with which a user can browse and discover songs by just listening to music and clicking a similar word sequence to jump to listen from there.

Figure 2 shows the LyricListPlayer screen. A music playlist is shown at top of the figure. The interface displays not only the lyrics of the song currently played back song (Fig. 2 top) but also its similar lyrics list (Fig. 2 bottom). The top of the list shows the lyrics of the song currently played back, and the other listed lyrics are ordered based on local similarity of latent topics. The list is called a "lyric list" in this paper. Hereafter, all lyrics in screenshots illustrated in this paper are taken from the RWC Music Database (Popular Music) [17]. Twenty songs (RWC-MDB-P Nos.81–100) are used as a playlist and the lyric list is also estimated from the songs.

LyricListPlayer provides the following three functions.

1) display music-synchronized lyrics
2) display music-synchronized lyric list
3) lyrics zapping interaction

2.1 Display music-synchronized lyrics and lyric list (similar word sequences)

The word currently being sung is highlighted by blue coloring so that, as in karaoke, a user can easily follow the current playback position. As its retrieved results, the similar word sequences are colored magenta (Fig. 3).

The range of coloring is determined by the length of the word sequence N (the number of morphemes or words) considered for computing local similarity. The local similarity of the lyrics of the lyric list is changed based on N and whether or not the ordering is considered. For exam-

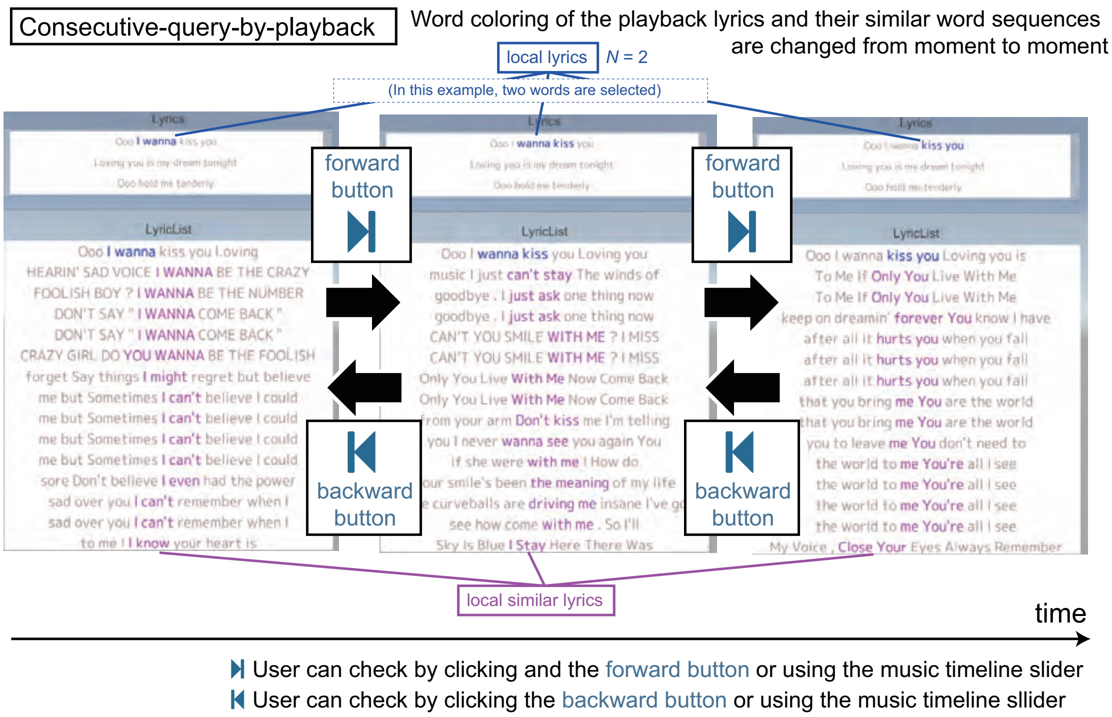

Figure 3. An example of changing playback position and similar word sequences along with the playback word.

ple, if the ordering is not considered, the word sequence "A B C" in a song has the same similarity as the sequence "A C B" in the same song.[3] Figure 4 and 5 show similar word sequences with different conditions: different N and with or without consideration of the words' ordering.

The coloring design is different from that used in the interface LyricSynchronizer and those used in well-known karaoke systems. LyricListPlayer colors the word (morpheme) currently sung and the subsequent $N-1$ words. The reason for this coloring design is to show information about both the current playback position' and the length currently used to compute local similarity'. To show the context of the similar word sequences in the lyric list, the previous and next words are also displayed.

The length of the word sequence (the number of words) and whether or not ordering is considered in computing similarity can be changed by using the two drop-down menus at the top of the screen (Fig. 2 top). The current playback position can be changed by using the forward and backward buttons to jump to the next/previous word (morpheme) or by using the music timeline slider.

2.2 Lyrics zapping interaction

The user can show the sung lyrics and their similar word sequences while listening to music. This is a kind of *passive* music information retrieval because the user does not input a query explicitly/actively.

On the other hand, the displayed similar word sequences can be clicked for zapping, to jump to listen from that point (Fig. 6). The zapping history is displayed at the right-side of the screen (Fig. 2) and can be clicked to back to a song played back before zapping. In addition, as a potential of the lyric list, the similar word sequences can be played back continuously to get an overview of the sequences sung by different artists or contexts.

[3] In the current implementation, the same word sequences of different songs have different similarities.

3. IMPLEMENTATION

The system first synchronizes the phoneme-level pronunciation of the lyrics with the musical audio signals for all the songs in the playlist. The estimated onset time and durations of all phonemes are converted to morpheme-level for Japanese lyrics and to word-level for English lyrics. This synchronization is called *lyrics alignment*.

Then, to compute similarity between words, the system estimates the latent topics of lyrics. Finally, the system calculates similarity among all word sequences with different N in the range $N = 1, 2, ..., 5$. The indexes of 200 word sequences having high similarity for each word sequence are stored for display on the lyric list screen. This interface support Japanese and English lyrics, and Japanese lyrics are spelled in a mixture of Japanese phonetic characters and Chinese characters.

3.1 Lyrics alignment

The phonetic-to-audio synchronization is estimated through Viterbi alignment with a phoneme-level hidden Markov model (monophone HMM) that is used as an acoustic model. We trained Japanese and English monophone HMMs by using the RWC Music Database (Popular Music) [17] with our own phonetic annotations; 80 Japanese songs are used to train a Japanese acoustic model and 20 English songs are used to train an English acoustic model. Here we refer this song set as the RWC

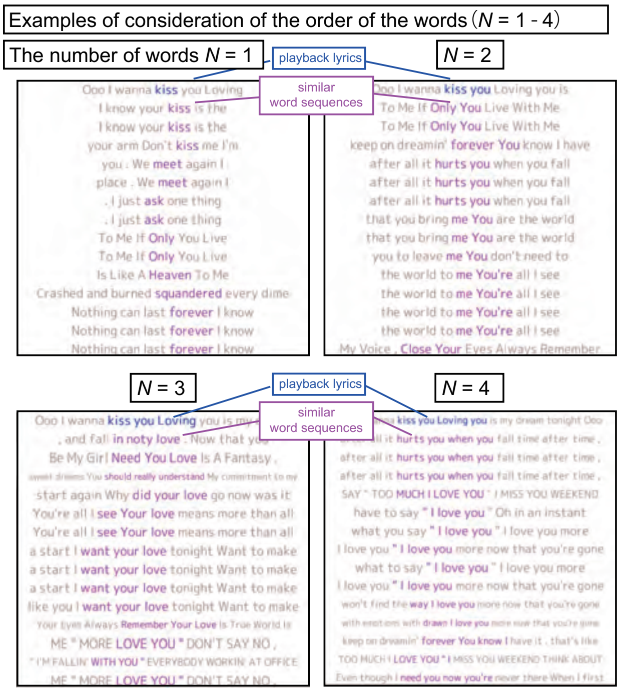

Figure 4. Examples of different N, the length of the word sequence (number of morphemes or words) for computing local similarity, with consideration of the ordering of the words.

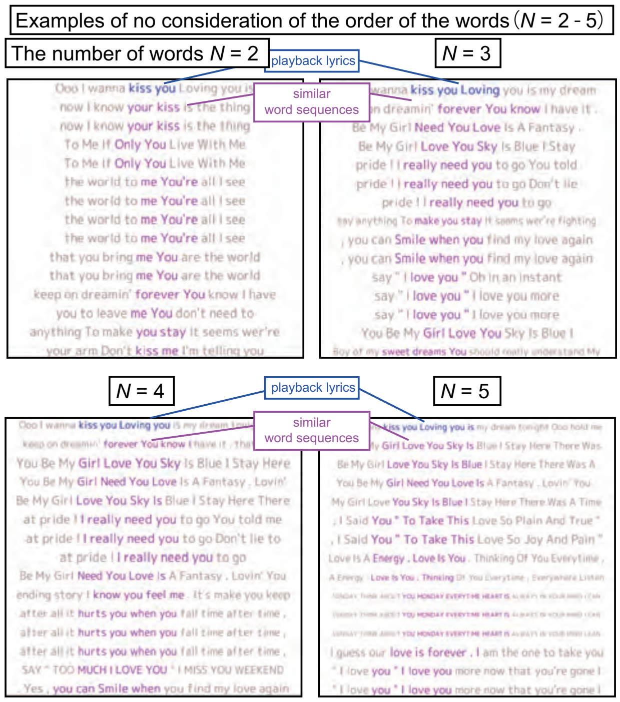

Figure 5. Examples by different N, the length of the word sequence (number of morphemes or words) for computing local similarity, with no consideration of the ordering of the words.

MDB.

To train the acoustic models, the pronunciation is estimated by using the Japanese language morphological analyzer MeCab [18] and the CMU pronouncing dictionary[4] for English lyrics. The acoustic features and alignment method are based on those used in LyricSynchronizer [19]. With regard to the acoustic features, we target monaural 16-kHz digital recordings and extract Δpower, 12th-order MFCCs, and 12th-order ΔMFCCs every 10 ms. To estimate the features, we performed separation of vocals from polyphonic musical audio signals [19].

3.2 Topic modeling

We use latent Dirichlet allocation (LDA) [20] for lyrics topic modeling. Since the LDA was originally proposed for text analysis, it can be used for lyrics modeling. In fact, there are three papers on work that used lyrics for LDA-based music retrieval [2, 7, 10]. The number of topics K is set to 100, and the model parameters of LDA are trained using the collapsed Gibbs sampler [21]. The conditions are based on previous work [7, 22].

The song set used for Japanese model training is 1,896 Japanese popular songs[5] and 80 lyrics of the RWC MDB. The Japanese popular songs appeared on a popular music chart in Japan[6] and were placed in the top twenty on weekly charts appearing between 2000 and 2008. The song set used for English model training is 2,314 English songs sung by 2,314 artists from Music Lyrics Database v.1.2.7[7], 51 English songs from commercial music CDs and 61 English lyrics of the RWC Music Database (20 from Popular Music, 10 from Royalty-Free Music, and 31 from Music Genre) [17, 23] are used.

For the topic modeling, all morphemes of Japanese are converted to the original form by using the MeCab for Japanese lyrics. Symbols such as punctuation marks and exclamation marks are used for model training because they can be used to express emotions or feelings. Finally, the vocabulary size in the 1,976 Japanese lyrics is 19,390 words (morphemes), and the vocabulary size in the 2,426 English lyrics is 23,756 words.

3.3 Similarity computing

By using a variational Bayesian inference of the LDA model training, the *responsibility*[8] (mixing weights) of multiple topics for each word can be estimated. Then the responsibilities of a word can be interpreted as the number of observations of the corresponding topic. To obtain responsibilities (unigram probabilities) for a set of words with the length N of 2 or more without consideration of the ordering, the word's responsibilities are summed (Fig. 7). Since this summing approach can be used to compute similarity between two sets of words with different N, it can also be used to compute similarity between two lines.

Since each topic can be represented by a unigram probability of the vocabulary, the distance between two words

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[5] Note that some are Western popular songs and English is used in them.

[6] http://www.oricon.co.jp/

[7] http://www.odditysoftware.com/page-datasales1.htm

[8] This term is from an article [24].

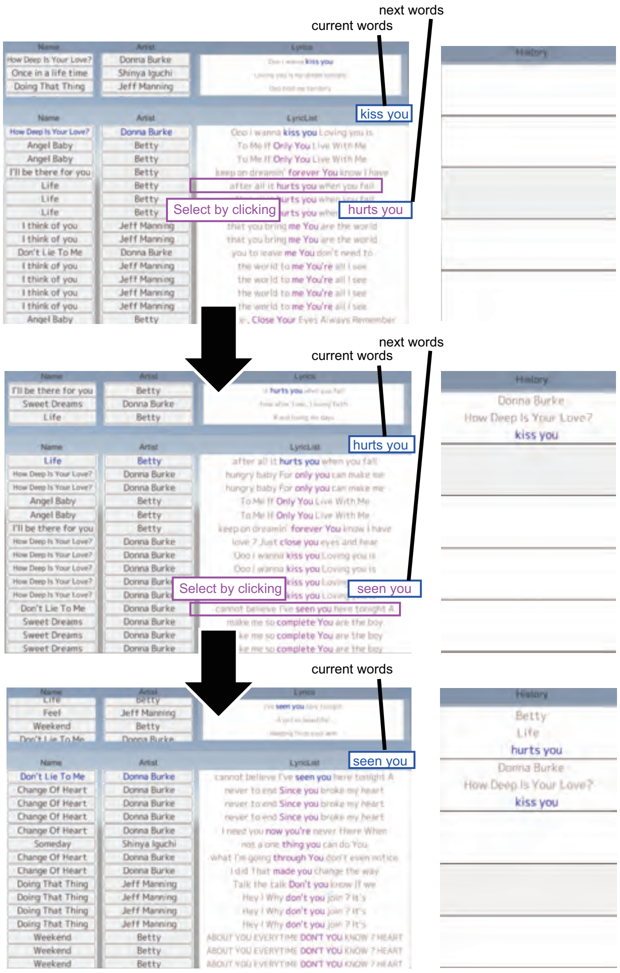

Figure 6. The local similar word sequences can be clicked to jump to listen from that point (lyrics zapping interaction).

is defined in this paper as the symmetric Kullback-Leibler distance (KL2) between two unigram probabilities.

To calculate similarity for words with the length N of 2 or more with consideration of the ordering, we first compute each word-pair similarity. In the current implementation, their median value is used as the similarity. For example, for two set of words, ABC and DEF, similarities A-D, B-E, and C-F are calculated first.

4. USER FEEDBACK

To investigate the capabilities, limitations, and potential of our interaction design, we asked eight users to use the system for 20 minutes and collected preliminary user feedback. We chose users, seven males and one female (U1–U8), who had different types of appreciation of music with lyrics. Five users had been conscious of lyrics while listening to music (U1, U2, U4, U5, and U8). Additionally, two user had occasionally chosen songs based on lyrics (U4 and U8).

The playlist consisted of the 10 Japanese songs, and we used the Japanese lyrics topic model. All users knew more than 1 song and six users knew 4 songs or more (U1, U2, U3, U4, U5, and U6). After the trial usage, we asked the

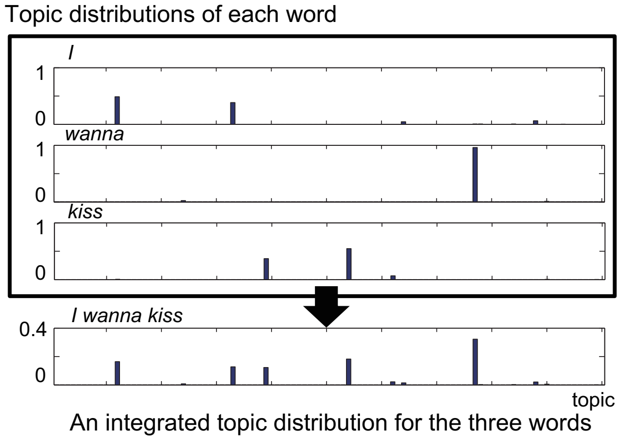

Figure 7. Examples of topic distributions and their integration.

users to write comments about two of the three primary functions of our system: 1) lyrics display (similar word sequences) and 2) lyrics zapping interaction.

Positive comments about the capabilities and potential of the interface were obtained from the users. Three users (U3, U4, and U7) used the function to display and listen to similar word sequences with enjoyment, and two users (U1 and U5) indicated that the function is helpful/useful for retrieving lyrics. One user (U6) indicated that the function changing the length of words for computing similarity was easy to use. One user (U3) frequently changed the length ($N = \{3, 4, 5\}$), and another user (U2) used only $N = 1$. With respect to the zapping interaction, three users (U2, U4 and U6) felt good about jumping/listening to the corresponding matching positions within the audio recordings.

Three users (U2, U6, and U7) indicated that LyricListPlayer is useful to be conscious of lyrics while listening to music. Furthermore, five users (U2, U5, U6, U7, and U8) indicated that a function zapping automatically is a way to expand possibility of application.

On the other hand, all users indicated that the speed with which the similar word sequences change should be controlled or be more effective and that either controlling it or making it more effective would be a good future direction for interface improvement. In association with that, four users (U2, U4, U6, and U8) thought that uncharacteristic words (*e.g.*, prepositions) should be omitted from the retrieving process. To improve practical utility of the system, five users (U2, U5, U6, U7, and U8) wanted to know a similarity between two songs based on lyrics or acoustic features (*e.g.*, mood, melody and musical structure).

5. CONCLUSION

This paper presents LyricListPlayer, a lyrics-synchronized music playback interface for retrieving lyrics passively. LyricListPlayer is also an active music listening interface based on lyrics, and an active listening style could help people be conscious of lyrics while listening to music. By taking into account the meaning of lyrics while listening, listeners can enrich their listening experience and become more emotionally involved with songs.

LyricListPlayer has an interactive function change the word length used for computing local similarity. Although there are works focusing on the similarity of music fragments or entire musical pieces [25] and the use of similarity by DJs connecting two pieces smoothly and for musical browsing [26, 27], to our knowledge, there is no research on how an interaction could be used to change the local range. Since music is a time-series media content, local similarity is an important aspect to deal with.

In future work, we plan to consider various word lengths for computing local similarity. We are also going to explore interactive designs for the display of similar word sequences, that is, to improve the interaction in ways based on user feedback. Although this interface focused only on lyrics-based information, information about other musical elements, such as vocal timbre and melody, should be integrated to enrich user experience. Moreover, a framework that can deal with a large amount of songs is also important to music listeners.

LyricListPlayer focused on an interaction design to explore "how to listen to a set of songs by using lyrics". The digitization of music and the distribution of content over the web have greatly increased the number of musical pieces available. Although music recommender systems and music information retrieval methods facilitate retrieving and listening to a large set of music, a recommended set of songs have to be listened to determine which song is one's favorite. Since the time one can spend listening to music is limited, more investigations of interactions for listening to a musical piece and/or a set of pieces are needed.

Acknowledgments

This paper utilized the RWC Music Database (Popular Music, Royalty-Free Music, and Music Genre). This work was supported in part by CREST, JST.

6. REFERENCES

[1] W. Machida and T. Itoh, "Lyricon: A visual music selection interface featuring multiple icons," in *Proc. IV 2011*, 2011, pp. 145–150.

[2] E. Brochu and N. de Freitas, ""Name That Song!": A probabilistic approach to querying on music and text," in *Proc. of NIPS2002*, 2002, pp. 1505–1512.

[3] M. Müller *et al.*, "Lyrics-based audio retrieval and multimodal navigation in music collections," in *Proc. ECDL'07*, 2007, pp. 112–123.

[4] R. Neumayer and A. Rauber, "Multi-modal music information retrieval – visualisation and evaluation of clusterings by both audio and lyrics," in *Proc. ISMIR 2007*, 2007.

[5] H. Fujihara *et al.*, "Hyperlinking Lyrics: A method for creating hyperlinks between phrases in song lyrics," in *Proc. ISMIR 2008*, 2008, pp. 281–286.

[6] D. Baur *et al.*, "SongWords: Exploring music collections through lyrics," in *Proc. ISMIR 2010*, 2010.

[7] S. Sasaki *et al.*, "LyricsRadar: A lyrics retrieval system based on latent topics of lyrics," in *Proc. ISMIR 2014*, 2014, pp. 585–590.

[8] R. Takahashi *et al.*, "Building and combining document and music spaces for music query-by-webpage system," in *Proc. of Interspeech 2008*, 2008, pp. 2020–2023.

[9] F. Kleedorfer, "Oh Oh Oh Whoah! towards automatic topic detection in song lyrics," in *Proc. ISMIR 2008*, 2008.

[10] L. Sterckx *et al.*, "Assessing quality of unsupervised topics in song lyrics," in *Advances in Information Retrieval, ECIR 2014*, 2014, pp. 547–552.

[11] B. Logan, "Semantic analysis of song lyrics," in *Proc. ICME 2004*, 2004.

[12] C. Laurier, "Multimodal music mood classification using audio and lyrics," in *Proc. ICMLA 2008*, 2008.

[13] C. McKay *et al.*, "Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features," in *Proc. ISMIR 2008*, 2008, pp. 213–218.

[14] D. J. Hu and L. K. Saul, "A probabilistic topic model for unsupervised learning of musical key-profiles," in *Proc. of ISMIR2009*, 2009, pp. 441–446.

[15] M. V. Zaanen and P. Kanters, "Automatic mood classification using TF*IDF based on lyrics," in *Proc. ISMIR 2010*, 2010, pp. 75–80.

[16] M. Goto, "Active music listening interfaces," in *Proc. ICASSP 2007*, 2007.

[17] M. Goto *et al.*, "RWC Music Database: Popular, classical, and jazz music databases," in *Proc. ISMIR 2002*, 2002, pp. 287–288.

[18] T. Kudo, "MeCab: Yet another part-of-speech and morphological analyzer," http://mecab.sourceforge.net/.

[19] H. Fujihara *et al.*, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[20] D. M. Blei *et al.*, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[21] T. L. Griffiths and M. Steyvers, "Finding scientific topics," in *Proc. Natl. Acad. Sci. USA (PNAS)*, vol. 1, 2004, pp. 5228–5235.

[22] T. Nakano *et al.*, "Musical similarity and commonness estimation based on probabilistic generative models," in *Proc. IEEE ISM 2015*, 2015.

[23] M. Goto *et al.*, "RWC Music Database: Music genre database and musical instrument sound database," in *Proc. of ISMIR 2003*, 2003, pp. 229–230.

[24] C. M. Bishop, *Pattern recognition and machine learning*. Springer-Verlag New York, Inc., 2006.

[25] M. Goto and K. Hirata, "Recent studies on music information processing," *Acoustical Science and Technology (edited by the Acoustical Society of Japan)*, vol. 25, no. 6, pp. 419–425, 2004.

[26] H. Ishizaki *et al.*, "Full-automatic DJ mixing with optimal tempo adjustment based on measurement function of user discomfort," in *Proc. ISMIR 2009*, 2009, pp. 135–140.

[27] T. Hirai *et al.*, "MusicMixer: Computer-aided DJ system based on an automatic song mixing," in *Proc. ACE 2015*, 2015, pp. 1–5.