HYPERLINKING LYRICS: A METHOD FOR CREATING HYPERLINKS BETWEEN PHRASES IN SONG LYRICS

Hiromasa Fujihara, **Masataka Goto**, and **Jun Ogata**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{h.fujihara, m.goto, jun.ogata} [at] aist.go.jp

ABSTRACT

We describe a novel method for creating a hyperlink from a phrase in the lyrics of a song to the same phrase in the lyrics of another song. This method can be applied to various applications, such as song clustering based on the meaning of the lyrics and a music playback interface that will enable a user to browse and discover songs on the basis of lyrics. Given a song database consisting of songs with their text lyrics and songs without their text lyrics, our method first extracts appropriate keywords (phrases) from the text lyrics without using audio signals. It then finds these keywords in audio signals by estimating the keywords' start and end times. Although the performance obtained in our experiments has room for improvement, the potential of this new approach is shown.

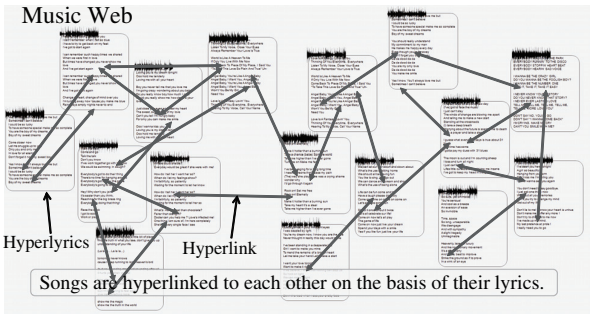

Figure 1. Hyperlinking lyrics. on Music Web

1 INTRODUCTION

The goal of this study is to enable a *Music Web* where songs are hyperlinked to each other on the basis of their lyrics (figure 1). Just as some hypertext phrases on the web are hyperlinked, so some phrases of lyrics (which we call *hyperlyrics*) on the Music Web can be hyperlinked. Such a hyperlinked structure of the Music Web can be used as a basis for various applications. For example, we can cluster songs based on the meanings of their lyrics by analyzing the hyperlinked structure or can show relevant information during music playback by analyzing the hyperlinked songs. We can also provide a new active music listening interface [1] where a user can browse and discover songs by clicking a hyperlinked phrase in the lyrics of a song to jump to the same phrase in the lyrics of another song. Although we can think of many possible ways to hyperlink musical pieces on the Music Web, focusing on song lyrics is natural because the lyrics are one of the most important elements of songs and often convey their essential messages.

Most approaches for analyzing inter-song relationship have been based on musical similarity between songs, and various music interfaces based on such song-level similarity have been proposed [2, 3, 4, 5, 6]. the methods of music browsing for intra-song navigation have also been studied, such as music browsing based on the structure [7, 8] and the lyrics [9, 10, 11]. However, hyperlinking lyrics — i.e., a combination of inter-song and intra-song navigations based on lyrics phrases — has not been proposed.

In this paper, we propose a method for hyperlinking identical keywords (phrases) that appear in the lyrics of different songs. Hyperlinking lyrics enables us to benefit from various studies dealing with sung lyrics in musical audio signals. For example, by using methods for automatic synchronization of lyrics with musical audio signals[9, 10], we can first find a keyword pair in the text lyrics for two different songs and then locate (i.e., estimate the start and end times of) the keyword in the audio signals of each song. However, it is still difficult to achieve accurate automatic lyrics recognition that enables us to find a keyword pair in sung lyrics that are recognized in polyphonic musical audio signals, despite the achievements made by studies on automatic lyrics recognition for musical audio signals [12, 13, 14]. While studies have also been done on analyzing text lyrics without using audio signals [15, 16], we cannot use their results to hyperlink lyrics.

Figure 2 shows the strategy our method uses to hyperlink lyrics. We assume that a user has a song database that is a set of audio files (e.g., MP3 files) and that we can prepare text files of the lyrics for some of the songs. Note that we do not require text lyrics for all the songs — i.e., a song database consists of songs with their text lyrics and songs without them. We therefore apply two different strategies for hyperlinking: one for hyperlinking from a phrase in other text lyrics to the same phrase in the text lyrics, and one for hyperlinking from a phrase in the text lyrics to the same phrase in the sung lyrics in a song (polyphonic music signals) without its text lyrics. Here, "text lyrics" means a text document containing the lyrics of a song, and "sung lyrics" means audio signals containing the lyrics sung by a singer in a polyphonic sound mixture. Although hyperlinking from text lyrics to text lyrics is relatively easy with the support of LyricSynchronizer [10, 1], hyperlinking from text lyrics to sung lyrics is more difficult.

2 OUR METHOD FOR HYPERLINKING LYRICS

Our method hyperlinks phrases that appear in the lyrics of different songs. In other words, if different songs share

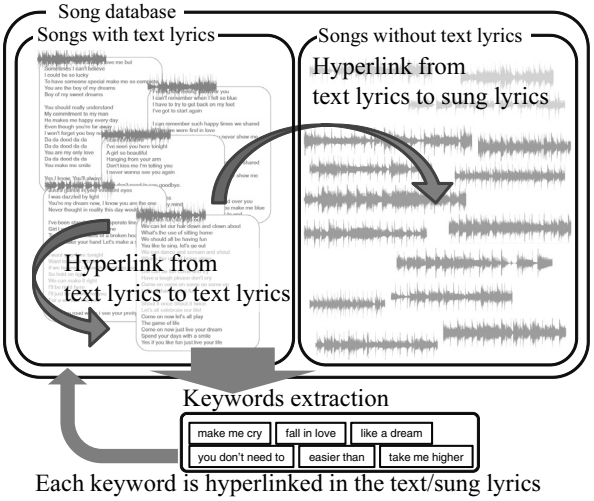

Figure 2. Two different strategies for hyperlinking: hyperlink from text lyrics to text lyrics and hyperlink from text lyircs to sung lyrics.

the same phrase (what we call a *keyword*[1]) in their lyrics, a section (temporal region) corresponding to the keyword in one song is hyperlinked with a section corresponding to the same keyword in another song. This method should deal with polyphonic music mixtures containing sounds of various instruments as well as singing voices. If a song database consists of songs with text lyrics and songs without text lyrics, this approach creates two different types of bidirectional hyperlinks: a hyperlink between songs with text lyrics and a hyperlink from a song with text lyrics to a song without them. The former hyperlink can be created by extracting potential keywords from all the text lyrics and finding sections corresponding to them (i.e., temporally locating them) in audio signals with the help of their lyrics. This estimation can be done using an automatic lyrics synchronization method described in [10]. For the latter hyperlink, our method looks for sections including voices that sing the keywords by using a keyword spotting technique for polyphonic music mixtures.

2.1 Hyperlinking from text lyrics to text lyrics

This section describes our method for hyperlinking from text lyrics to text lyrics. By using text lyrics of all the songs, the method first extracts as many keywords that can result in meaningful hyperlinks as possible. It then estimates sections (the start and end times) of each keyword in audio signals. Finally, it creates bidirectional hyperlinks between the estimated keyword sections.

2.1.1 Keyword extraction from the text lyrics

We would like to create as many hyperlinks as possible while ensuring that they are meaningful and useful. We therefore have to accordingly extract keywords. From the viewpoint of inter-song hyperlinks, each keyword must appear in multiple songs (the more songs, the better). In addition, since long keywords tend to convey important meanings, longer keyword lengths are preferred. Longer keywords are also advantageous for improving the accuracy of keyword detection, in Sec. 2.2. In contrast, short words/phrases, such as an article and a preposition, are not appropriate as keywords.

The requirements for appropriate keywords can be summarized as follows:

(a) A larger number of songs sharing a keyword is better.
(b) A larger number of phonemes in a keyword is better.

Note the trade-off between these requirements: a longer phrase is less likely to appear many times. We therefore try to maximize the number of phonemes provided each keyword appears in two songs or more.

According to these requirements, we developed the keyword extraction algorithm shown below.

1. Initialize a keyword dictionary as empty. Each keyword registered to this dictionary in the following will have a flag called a "finalized" flag only when it cannot be connected with an adjacent word to make a longer keyword (phrase).
2. Register every word of all text lyrics to the keyword dictionary.
3. Count the number of different songs whose lyrics include each keyword of the keyword dictionary. The keyword with the largest number of corresponding songs is considered the most frequent keyword.
4. Remove all keywords that appear in less than M songs from the keyword dictionary.
5. Select the most frequently occurring keyword without the "finalized" flag. If all the keywords have this flag, this algorithm is finished.
6. Try to combine adjacent words with the selected keyword. The former word and the latter word of each lyrics are respectively combined to make a longer phrase.
7. Only the best combination that results in the most frequent phrase is registered as a new keyword to the keyword dictionary, provided it appears in M songs or more. Note that the original keyword is not removed even if a combined one is thus registered.
8. If any combination for the selected keyword does not appear in M songs or more, the "finalized" flag is attached to the keyword when the number of phoneme of it is more than "N" and the keyword is removed when the number of phoneme of it is under "N".
9. Go back to 5.

First, the algorithm uses a dictionary to prepare the pronunciation of each word of all text lyrics so that each word can be represented as a sequence of phonemes and the number of phonemes can be calculated. [2]

Two parameters, M and N, correspond to the above requirements. Since their appropriate values will depend on the total volume of all lyrics, in Sec. 2.2 we discuss how they are set.

[1] In this paper, the term "keyword" means a lyrics phrase consisting of one or more consecutive words.

[2] For songs sung in the Japanese language, we need to apply a preprocessing, called morphological analysis, so that text lyrics can be divided into a sequence of words. Unlike English, a word boundary is not explicitly specified by a space character in Japanese.

2.1.2 Hyperlinking keywords

Each keyword in the keyword dictionary is used to create bidirectional hyperlinks. Given the text lyrics, our method can easily find all the positions of each keyword through a string search. At each found position, the method then finds the keyword's section (temporal region) by estimating its start and end times in the corresponding musical audio signals. This can be done by using an automatic lyrics synchronization method [10] which estimates the start and end times of all words in the lyrics, including the keywords.

2.2 Hyperlinking from text lyrics to sung lyrics

This section describes our method for hyperlinking from songs with text lyrics to songs with only the sung lyrics (audio signals). Each keyword in the keyword dictionary is again used to create bidirectional hyperlinks, but our method should find each keyword in polyphonic musical audio signals without text lyrics. The method first judges whether the sung lyrics of each song (without text lyrics) includes a keyword, and then finds the keyword section by estimating its start and end times. We enable this through a method based on a keyword spotting technique [17]. This technique uses two different statistical acoustic models: a model for the keyword pronunciation (called the *keyword model*) and a model for other sounds (called the *garbage model*). By finding the best matching of these models to singing voices segregated from polyphonic audio signals, we can find keyword sections described by the keyword model. More specifically, we first detect many candidate sections of keywords and then narrow the candidates down by rescoring them.

2.2.1 Feature extraction for singing voices in polyphonic mixtures

To use the acoustic models, we extract a sequence of feature vectors from each song without the text lyrics. Since the audio signals are polyphonic, we have to reduce various influences of the accompaniment sounds to extract feature vectors that represents only the singing voice. We therefore use the feature extraction method described in [10]. This method estimates and resynthesizes the singing voice (the sung melody) in polyphonic mixtures through three steps:

(1) estimate the most predominant F0 as the melody line (singing voice candidate) by using the *PreFEst* method [18]

(2) extract the harmonic structure corresponding to the estimated F0

(3) use sinusoidal synthesis of the harmonics to resynthesize the audio signal (waveform) corresponding to the melody line.

After resynthesizing the singing voice, we extract the MFCCs, ΔMFCCs, and Δ Power as feature vectors.

2.2.2 Preparing acoustic models by training phone models

The keyword and garbage models are represented as the hidden Markov models (HMMs). In the keyword HMMs, the pronunciation of each keyword is represented as a sequence of phone models as shown in Figure 3. The garbage HMM is defined as a phoneme typewriter in which any phonemes can appear in any order as shown in Figure 4. The keyword HMMs and garbage HMM are integrated in parallel as shown in Figure 5.

The phone models that represent acoustic characteristics of the phonemes of singing voices significantly affect the performance of keyword spotting. Because we cannot directly use phone models for typical automatic speech recognition, we built from scratch our own phone models for singing voices. We prepared precise phoneme-level annotation for 79 Japanese songs of the RWC Music Database: Popular Music (RWC-MDB-P-2001) [19], and then trained monophone models with 3-state left-to-right HMMs on those songs.

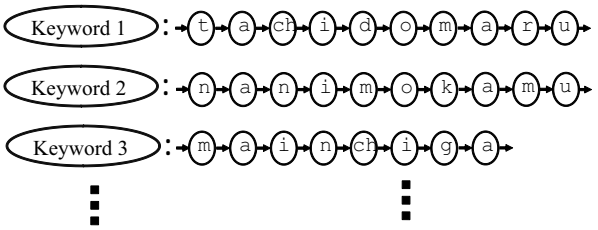

Figure 3. Keyword models (HMMs).

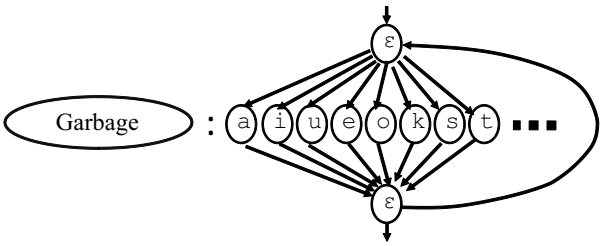

Figure 4. Garbage model (HMM).

2.2.3 Keyword candidate detection and score calculation

For each song without the text lyrics, the method detects candidate sections of each keyword by applying a Viterbi decoder to the feature vectors. As a result, the keyword HMMs can be expected to match with candidate sections where the singer sings the keyword, while the garbage HMM can be expected to match with all other sections. In this decoding, the likelihood of each candidate section against the corresponding keyword HMM can also be obtained, which represents acoustic matching between singing voices and the HMM. This decoding needs to use a word insertion penalty; each keyword in the keyword HMMs and each phoneme in the garbage HMM is regarded as a word. The word insertion penalty prevents long keywords from being substituted by short keywords of other keyword HMMs or phonemes of the garbage HMM. This decoding framework is based on typical automatic speech recognition techniques, except that the language model (Figure 5) was designed specifically to detect the keywords.

After the candidate sections of keywords are detected, we calculate the score of each candidate to narrow down the number of candidates. First, the Viterbi decoder using only the garbage HMM is applied to each candidate section to obtain its likelihood against the garbage HMM. Since the likelihood of each candidate section against the corresponding keyword HMM has already been obtained as explained above the score can be defined as the difference in the average log likelihood between the keyword HMM and the

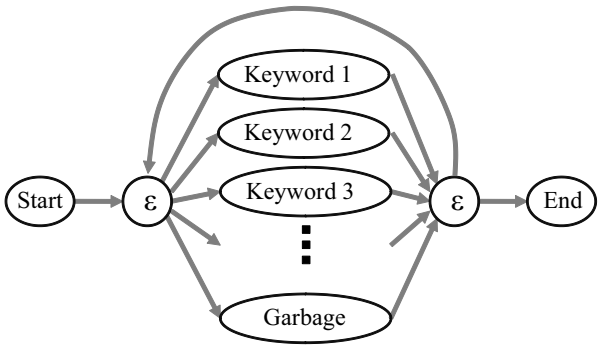

Figure 5. Integration of the keyword models and the garbage model.

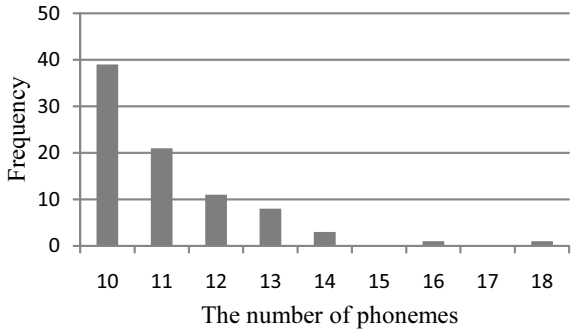

Figure 6. Distribution of the number of phonemes of extracted keywords.

garbage HMM.

2.2.4 Hyperlinking keywords

For each keyword, we select several songs that include detected keyword candidate sections with high scores. We then create bidirectional hyperlinks from keyword sections in songs with the text lyrics to the selected candidate sections in songs without the text lyrics.

3 EXPERIMENTS

We conducted preliminary experiments to evaluate our hyperlinking method by using 79 Japanese songs taken from the RWC Music Database: Popular Music (RWC-MDB-P-2001) [19].

3.1 Evaluation of hyperlinking lyrics

First, we evaluated the performance of the keyword extraction described in Sec. 2.1.1. We set parameters M and N to 2 (songs) and 10 (phonemes), respectively. As a result, 84 keywords were automatically extracted and the average number of phonemes of those keywords was 11.1. [3] Figure 6 shows a distribution of the number of phonemes and Table 1 shows examples of the extracted keywords. We found that appropriate phrases that could result in meaningful hyperlinks were chosen as the keywords.

Next, we evaluated the performance of hyperlinking from text lyrics to sung lyrics, as described in Sec. 2.2 by introducing a new evaluation measure *link success rate* [4] . The link success rate indicates the percentage of hyperlinks that correctly connected a phrase in a song to the same phrase appearing in another song. Our evaluation procedures were as follow:

1. Assuming that the lyrics of the 79 songs were unknown, we executed the keyword candidate detection and score calculation described in Sec. 2.2.3.
2. For each song in the 79 songs, we again assumed that we obtained the text lyrics of this song only and created the hyperlinks of the above 84 keywords from this song to the other 78 songs. In this experiment, the hyperlinks were created only for the keyword candidate section with the highest score.
3. We evaluated these hyperlinks by using the link success rate and averaged the rate over all 79 songs.

The link success rate, r, of a song is expressed as

$$r = \frac{\sum_{k=1}^{K} \sum_{i=1}^{I_k} s(w(k,i))}{\sum_{k=1}^{K} I_k}, \quad (1)$$

$$s(w(k,i)) = \begin{cases} 1 & \text{if } w(k,i) \text{ is appropriate} \\ 0 & \text{if } w(k,i) \text{ is not appropriate} \end{cases}, \quad (2)$$

where k denotes a different keyword, K denotes the number of keywords appearing in the song, I_k denotes the number of occurences of the k-th keyword within this song (note that the same keyword is sometimes repeated in the same song), and $w(k,i)$ expresses the i-th hyperlink of the k-th keyword. A hyperlink $w(k,i)$ is judged to be appropriate when more than half of the detected region (in the other 78 songs) hyperlinked from the k-th keyword overlaps with the ground truth region (given by the annotation we used in Sec. 2.2.2. The experimental result showed that the link success rate was 30.1%.

3.2 Number of phonemes and occurrences of keywords

We then evaluated the appropriateness of parameters M (= 2 songs) and N (= 10 phonemes). As described in Sec. 2.1.1, M indicates the minimum number of songs in which each keyword should appear and N indicates the minimum number of phonemes of each keyword (i.e., the minimum length of each keyword). Because we would like to create as many hyperlinks as possible, we first fixed M as 2 songs for this evaluation with a small number of songs. We then measured the link success rate by increasing N. Table 2 shows the dependence of the link success rate and the number of extracted keywords on the number of phonemes in the keywords. Taking the trade-off between the two parameters into consideration, we set N as 10 phonemes.

3.3 Validation of our experimental conditions

In the experiments described in Secs. 3.1 and 3.2, the 79 songs used for the evaluation of hyperlinks were the

[3] Since the lyrics used here were written in Japanese, we used a preprocessing morphological analyzer MeCab[20] to divide the input text lyrics into word sequences.

[4] In this paper, we have omitted any evaluation of the hyperlinking from text lyrics to text lyrics, described in Sec. 2.1, because its accuracy depends on only the performance of LyricSynchronizer [10, 1]

Table 1. Examples of extracted keywords. Note that English translations were done specifically are prepared just for this paper because most keywords are not complete sentences and are hard to translate.

Keyword	English translation	Phoneme sequence	number of occurrences
が教えてくれたこと	what 〜 taught me	g a o sh i e t e k u r e t a k o t o	2 songs
どこまでも続く	continue forever	d o k o m a d e m o ts u z u k u	2 songs
心の中	in one's heart	k o k o r o n o n a k a	3 songs
素敵な笑顔	nice smile	s u t e k i n a e g a o	2 songs
世界中に	all over the world	s e k a i jy u: n i	2 songs

Table 2. Dependence of the link success rate on the number of phonemes.

# of phonemes	8	9	10	11	12	13
Link success rate (%)	23.2	27.5	30.1	24.3	35.9	40.0
# of keywords	271	144	84	45	24	13

Table 3. Results of phoneme-level recognition: Comparison between closed and open conditions.

Condition	i. closed	ii. open
Accuracy	50.9%	49.8%

same as those used for the phone model training. This was done because we had to apply the laborious time-consuming phoneme-level annotation on the 79 songs for both training the phone model and preparing the ground truth for the evaluation. Moreover, since the likelihood comparison between different songs requires use of the same phone model, we could not prepare different phone models by omitting the target song in the training and conduct a typical cross-validation. Since we used 79 songs for the training, we expected little contribution from the target song (used in the evaluation). Still, we wanted to confirm that this would not be a major issue affecting the reliability of our evaluation.

We therefore evaluated the performance of phoneme-level recognition using the Viterbi decoder and the phoneme typewriter under the condition that the text lyrics were not known. As the ground truth data for measuring the average frame level accuracy, we used the phoneme-level annotation used to train the phone model. Two two conditions were compared:

i. The same 79 songs were used for both the phone model training and the evaluation (closed condition).

ii. 10-fold cross validation was done on the 79 songs for the training and the evaluation (open condition).

Note that these conditions, the purpose, and the evaluation measure differ from those in Secs. 3.1 and 3.2.

Table 3 shows the results. These conditions did not greatly affect accuracy, confirming the appropriateness of our experimental conditions.

3.4 Evaluation of phone models

Finally, we compared the phone models trained from scratch with the phone models adapted from those for speech recognition. In [10], the phone models were prepared by adapting phone models for speech recognition with a small number of training data (10 songs with the phoneme-level annotation). They performed well for the lyrics synchronization problem, but the keyword detection problem, which is more difficult, requires more accurate phone models. We therefore prepared precise phoneme-level annotation and trained the phone models from scratch. To confirm the effectiveness of training of the phone models from a large amuont of training data, we conducted experiments using three phone models:

i. (small adaptation) Phone models created by adapting the phone model for speech recognition using 10 songs.

ii. (large adaptation) Phone models created by adapting the phone model for speech recognition using 79 songs.

iii. (training) Phone models trained from scrach using 79 songs.

The results are shown in Table 4. We found that the averaged frame level accuracy was drastically improved under the condition iii. This indicates that, when we have enough training data, phone models trained from scratch can perform better than the adapted ones based on speech recognition.

Table 4. Results of phoneme-level recognition: Comparison of three phone models.

	i. small adapt.	ii. large adapt.	iii. train.
Accuracy	27.1%	32.7%	50.9%

4 DISCUSSION

In this paper, we looked at the way of hyperlinking lyrics, i.e., creating bidirectional hyperlinks connecting lyrics keywords shared by different songs. Thus, we can generate a hyperlinked (networked) structure of songs. Our technology can be used when we only know the lyrics of a part of the songs. This step towards enabling the Music Web should help lead to a lyrics-based-MIR from songs that have unknown lyrics.

We can incorporate this method into *LyricSynchronizer* [10, 1], which displays scrolling lyrics with the phrase currently being sung highlighted during playback. A user interested in a different song containing the same hyperlinked keyword of the song currently being played can simply click on a highlighted keyword to jump to and listen from that keyword in a different song.

This is just the beginning for this new research topic, and because dealing with lyrics of polyphonic music signals is challenging, the performance represented by the experimental results given in Sec. 3.1 still need to be im-

proved. We expect to improve the overall performance by refining the phone model. As shown in Sec. 3.4, we can improve it by preparing the annotation for more songs. Moreover, as speech recognition technologies have shown, once we obtain good initial phone models, we can improve them through transcribing audio signals (i.e., lyrics) without requiring precise phoneme-level annotation.

Our experience in developing the keyword extraction method described in Sec. 2.1.1 has shown that the well known tf-idf (term frequency-inverse document frequency) is not useful for extracting keywords that have as many phonemes as possible but appear in two songs or more. The tf-idf is designed based of the assumptions that the importance of a keyword for a document is proportional to the number of occurrences of the keyword in the document is and inversely proportional to the number of occurrences of the keyword in all documents. However, the number of keyword occurrences in a song in our problem is unimportant. Even if a keyword appears only once, a user can jump from that keyword to other songs of interest. Therefore, we developed our own keyword extraction method.

5 CONCLUSION

This paper described a method for creating hyperlinks between different songs at the level of lyrics phrases. We presented a method for dealing with an important problem that has received little attention. We created two kinds of hyperlinks: hyperlinks between songs where the text lyrics are known and those from songs with text lyrics to songs without text lyrics. We created these hyperlinks by extracting keywords from the text lyrics of all songs and by detecting the keywords in polyphonic audio signals by using a HMM-based keyword spotting technique. Our experimental results show that the approach is promising, but better performance is needed for practical application. We will apply our method to a much larger database in the future. We also plan to develop useful applications for this method, including song clustering based on the songs' lyrics, and a Music Web browser that enables users to browse and discover interesting songs by clicking the *hyperlyrics* of songs.

6 ACKNOWLEDGEMENTS

This work was partially supported by CrestMuse, CREST, JST. We used the HTK (Hidden Markov Model Toolkit) [21] for training the phone models. We thank Katunobu Itou (Hosei University) for helping us to create the phoneme label annotation of the RWC Music Database [19]. We also thank everyone who has contributed to building and distributing the atabase.

7 REFERENCES

[1] Masataka Goto, "Active music listening interfaces based on signal processing," in *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, 2007, pp. IV–1441–1444.

[2] George Tzanetakis and Perry Cook, "MARSYAS: A framework for audio analysis," *Organised Sound*, vol. 4, no. 30, pp. 169–175, 1999.

[3] Robert Neumayer, Michael Dittenbach, and Andreas Rauber, "Playsom and pocketsomplayer: Alternative interfaces to large music collections," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, pp. 618–213.

[4] Takayuki Goto and Masataka Goto, "Musicream: New music playback interface for streaming, sticking, sorting, and recalling musical pieces," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, pp. 404–411.

[5] Elias Pampalk and Masataka Goto, "Musicrainbow: A new user interface to discover artists using audio-based similarity and web-based labeling," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006, pp. 367–370.

[6] Paul Lamere and Douglas Eck, "Using 3D visualizations to explore and discover music," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007, pp. 173–174.

[7] Masataka Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1783–1794, 2006.

[8] Meinard Müller and Frank Kurth, "Enhancing similarity matrices for music audio analysis," in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2006)*, 2006, pp. V–9–12.

[9] Ye Wang, Min-Yen Kan, Tin Lay Nwe, Arun Shenoy, and Jun Yin, "Lyrically: Automatic synchronization of acoustic musical signals and textual lyrics," in *Proceedings of the 12th ACM International Conference on Multimedia*, 2004, pp. 212–219.

[10] Hiromasa Fujihara, Masataka Goto, Ogata Jun, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno, "Automatic synchronization between lyrics and music CD recordings based on viterbi alignment of segregated vocal signals," in *Proceedings of the IEEE International Symposium on Multimedia (ISM 2006)*, 2006, pp. 257–264.

[11] Meinard Müller, Frank Kurth, David Damm, Christian Fremerey, and Michael Clausen, "Lyrics-based audio retrieval and multimodal navigation in music collections," in *Proceedings of the 11th European Conference on Digital Libraries (ECDL 2007)*, 2007.

[12] Chong-Kai Wang, Ren-Yuan Lyu, and Yuang-Chin Chiang, "An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker," in *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech2003)*, 2003, pp. 1197–1200.

[13] Toru Hosoya, Motoyuki Suzuki, Akinori Ito, and Shozo Makino, "Lyrics recognition from a singing voice based on finite state automaton for music information retrieval," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, pp. 532–535.

[14] Matthias Gruhne, Konstantin Schmidt, and Christian Dittmar, "Phoneme recognition in popular music," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007, pp. 369–370.

[15] Peter Knees, Markus Schedl, and Gerhard Widmer, "Multiple lyrics alignment: Automatic retrieval of song lyrics," in *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, pp. 564–569.

[16] Bin Wei, Chengliang Zhang, and Mitsunori Ogihara, "Keyword generation for lyrics," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007, pp. 121–122.

[17] Kate Knill and Steve Young, "Speaker dependent keyword spotting for accessing stored speech," Tech. Rep. CUED/F-INFENG/TR 193, Cambridge University, 1994.

[18] Masataka Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[19] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "RWC Music Database: Popular, classical, and jazz music databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, 2002, pp. 287–288.

[20] MeCab: Yet Another Part of Speech and Morphological Analyzer, "http://mecab.sourceforge.net/," .

[21] HTK: The Hidden Markov Model Toolkit, "http://htk.eng.cam.ac.uk/," .