A CHORUS-SECTION DETECTION METHOD FOR LYRICS TEXT

Kento Watanabe **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{kento.watanabe, m.goto}@aist.go.jp

ABSTRACT

This paper addresses the novel task of detecting chorus sections in English and Japanese lyrics text. Although chorus-section detection using audio signals has been studied, whether chorus sections can be detected from text-only lyrics is an open issue. Another open issue is whether patterns of repeating lyric lines such as those appearing in chorus sections depend on language. To investigate these issues, we propose a neural network-based model for sequence labeling. It can learn phrase repetition and linguistic features to detect chorus sections in lyrics text. It is, however, difficult to train this model since there was no dataset of lyrics with chorus-section annotations as there was no prior work on this task. We therefore generate a large amount of training data with such annotations by leveraging pairs of musical audio signals and their corresponding manually time-aligned lyrics; we first automatically detect chorus sections from the audio signals and then use their temporal positions to transfer them to the line-level chorus-section annotations for the lyrics. Experimental results show that the proposed model with the generated data contributes to detecting the chorus sections, that the model trained on Japanese lyrics can detect chorus sections surprisingly well in English lyrics and that patterns of repeating lyric lines are language-independent.

1. INTRODUCTION

The digitization of lyrics collections has opened various areas of lyrics-based research in the Music Information Retrieval (MIR) community, such as research on lyrics browsing [1–3], lyrics genre classification [4–6] and lyrics-to-audio synchronization [7–17]. Lyrics are usually plain text without any annotations, and some researchers have analyzed their structure, such as paragraph structure and topic transitions between paragraphs [18–22]. For example, Fell et al. [18] and Watanabe et al. [19] estimated section boundaries in lyrics text without empty lines but were not able to assign a section label such as verse or chorus to each estimated section. Chorus sections were not detected in lyrics text.

The goal of this paper is to achieve automatic chorus-section detection for lyrics text. This task has not been studied, though chorus-section detection, as well as music structure analysis, for audio signals has been a popular topic of research in the MIR community [23–41]. Since whether chorus sections can be detected from text-only lyrics is an open issue, it is worth investigating this issue from academic viewpoints. Moreover, a chorus-section detection method for lyrics text has potential applications. For example, when listeners want to find lyrics with a chorus section having a particular phrase such as "I love you" for the purpose of singing, reusing its chorus section in a short video clip, etc., it is necessary for a lyrics search system to automatically detect which lines of the lyrics are included in chorus sections. The detected lyric lines of chorus sections could be used in a lyrics viewing function of music services displaying lyrics with those lines highlighted by a different color or typeface. Automatic lyric video generation technologies could give those lines more vivid animations.

Chorus sections are the most repeated and memorable portions of a song [39]. Since it is not easy to explore heuristic rules to find such sections, most existing chorus-section detection methods for audio signals have leveraged repetitive patterns of those sections within a song. In this paper, we propose a supervised model that can detect chorus sections in English and Japanese lyrics. Our model uses both structural features that represent patterns of repeating lyric lines and linguistic features that are calculated from word2vec [42] and context2vec [43]. To detect chorus sections using only plain text without any labels or even empty lines (i.e., section boundaries), we investigate a model and features effective for chorus-section detection. Experimental results show that our proposed model outperforms alternative baseline models and that combining structural and linguistic features contributes to better performance.

Although such a supervised model needs a large dataset of lyrics with line-level chorus-section annotations for its training, there was no such dataset as there was no prior work on chorus sections in lyrics text. To address this issue of lacking training data, we generated a dataset consisting of 9,313 English and 91,459 Japanese lyrics with chorus-section annotations by utilizing pairs of musical audio signals and their corresponding manually time-aligned lyrics. We first automatically detected chorus sections in audio signals of a song [39]. Then, since each lyric line had the corresponding start time within the song, we could find lyric lines that temporally correspond to the duration of each detected chorus section. We thus obtained the annotated dataset by assigning a `chorus` label to those lyric

	Text (a sequence of lines x_t in lyrics)	Label y_t
1:	ooo I wanna kiss you	chorus (1)
2:	loving you is my dream tonight	chorus (1)
3:	ooo hold me tenderly	chorus (1)
4:	loving me with all your heart	chorus (1)
5:	boy you never tell me that you love me	not-chorus (0)
6:	I'm going crazy wondering about you baby	not-chorus (0)
7:	do you really know boy how much I care	not-chorus (0)
8:	could you really show me how deep is your love ?	not-chorus (0)
9:	just close you eyes and hear my heart	not-chorus (0)
10:	the sweet sweet beat of my love	not-chorus (0)
11:	can't you tell I'm hungry baby	not-chorus (0)
12:	for only you can make me smile	not-chorus (0)
13:	ooo I wanna kiss you	chorus (1)
14:	loving you is my dream tonight	chorus (1)
15:	ooo hold me tenderly	chorus (1)
16:	loving me with all your heart	chorus (1)
17:	can't you understand me my point of view	not-chorus (0)
18:	do you really love me beyond all words	not-chorus (0)
19:	I just need to hear now from your sweet lips	not-chorus (0)
20:	I'm the only girl you ever want to kiss	not-chorus (0)
21:	just close your eyes and hear my heart	not-chorus (0)
22:	the sweet sweet beat of my love	not-chorus (0)
23:	can't you tell I'm hungry baby	not-chorus (0)
24:	for only you can make me smile	not-chorus (0)
25:	just close your eyes and hear my heart	not-chorus (0)
26:	the sweet sweet beat of my love	not-chorus (0)
27:	can't you tell I'm hungry baby	not-chorus (0)
28:	for only you can make me smile	not-chorus (0)
29:	ooo I wanna kiss you	chorus (1)
30:	loving you is my dream tonight	chorus (1)
31:	ooo hold me tenderly	chorus (1)
32:	loving me with all your heart	chorus (1)

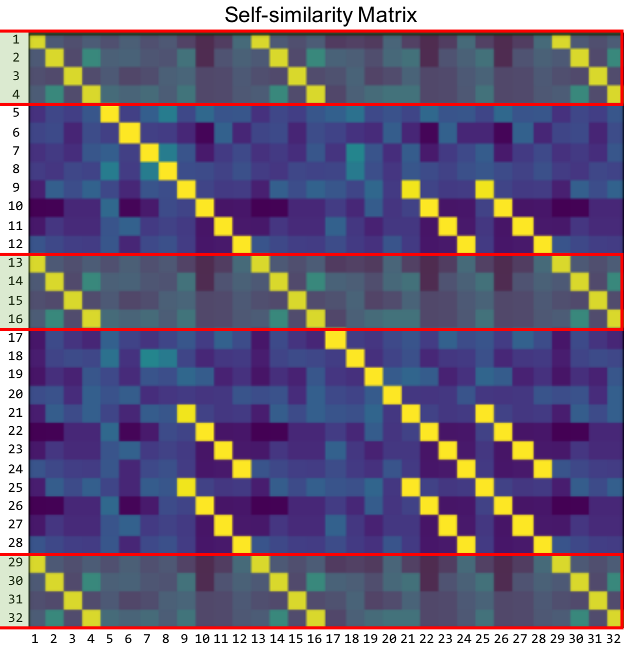

Figure 1. Example of lyrics with chorus-section annotations and corresponding self-similarity matrix in which each cell represents the similarity between two lyric lines. These lyrics are from "How Deep Is Your Love?" (RWC-MDB-P-2001 No. 81 in the RWC Music Database [44]).

lines and a `not-chorus` label to the other lines. Experimental results show that the model trained with this large automatically generated dataset performs better than the model trained with a smaller manually annotated dataset and that the model trained on Japanese lyrics can detect chorus sections surprisingly well in English lyrics.

2. LYRICS CHORUS-SECTION DETECTION TASK

The left side of Figure 1 shows an example of lyrics with chorus-section annotations (labels). The lyrics of a song are a sequence of lyric lines, each line having a sentence or phrase. In this example there are three highlighted chorus sections that have exactly the same four lines, though in other songs, lyrics of chorus sections are repeated with some modifications. To maximize the applicability, as shown in this example, we assume that the input text of lyrics does not have any section boundaries. Even though some lyrics contain empty lines at those boundaries, those lines are deleted in advance. We also assume that the input text does not have explicit chorus labels such as "(chorus)" at the beginnings of chorus sections. Even though some lyrics contain those labels, they are deleted as well. When lyrics contain a repetition label such as "(* repeat)", it is manually replaced with the corresponding lyric lines.

We formulate this chorus-section detection task as a sequence labeling problem: predicting the `chorus` or `not-chorus` status for each lyric line. Let X_s be the lyrics of a song s composed of T lines of text: $X_s = \{x_1, ..., x_t, ..., x_T\}$. Each lyric line x_t has a binary label y_t. If $y_t = 1$, x_t is in a chorus section. If $y_t = 0$, x_t is not in a chorus section. Y_s denotes a sequence of labels corresponding to X_s: $Y_s = \{y_1, ..., y_t, ..., y_T\}$. In the training step, the model learns the conditional probability $P(Y_s|X_s)$. In the validation/testing step, the trained model has to predict labels Y_s for given lyric lines X_s.

Chorus sections cannot be detected by simply extracting repeated lines since those lines often correspond to non-chorus sections. For example, lyric lines 9–12 and 21–24 in Figure 1 are exactly repeated, but those lines are not in chorus sections. It is also difficult to manually define a set of rules to find various chorus sections. We therefore prepare various features that could be useful for machine learning to deal with various types of chorus sections.

3. COMPUTATIONAL MODELING OF CHORUS SECTIONS IN LYRICS

We propose a neural network-based model for sequence labeling by using structural features that are self-similarity matrix (SSM) representations. SSM representations are widely used in computational music structure analysis, but we use different representations for lyrics. In addition to structural features, our model utilizes linguistic features such as word vectors and sentence vectors calculated from word2vec [42] and context2vec [43], which are widely used in natural language processing.

In the following sections, we first describe nine SSMs for capturing patterns of repeating lyric lines and explain how to encode the SSMs for neural networks (Section 3.1). We then describe the linguistic features obtained by vectorizing the semantic/syntactic information of lines using word2vec and context2vec (Section 3.2). Finally, we describe a neural-network-based sequence labeling model with these structural and linguistic features (Section 3.3).

3.1 Structural Features

Most previous work on music structure analysis for audio signals [23–41] identifies repeated musical sections by using a SSM like that shown in Figure 1. Repeated sections lead to high values in diagonals of the matrix, and those patterns are used to identify the structure. To capture repeated lyric lines that often appear in chorus sections, we also compute the SSM from lyrics text, but the design of the similarity measure to compute each cell of the SSM is important. We propose to use the following nine variations of similarity measures sim_m, where m denotes the variation. Some of the similarities are based on previous studies [18, 19].

String similarity (sim_{str}): a normalized Levenshtein edit distance [45] between the characters of two lyric lines.

Head similarity (sim_{head}): a normalized Levenshtein edit distance between the characters of the first two words of two lyric lines.

Tail similarity (sim_{tail}): a normalized Levenshtein edit distance between the characters of the last two words of two lyric lines.

Phonetic similarity (sim_{phone}): To capture rhymes in the lyrics, we calculate a normalized Levenshtein edit distance between the phonetic transcriptions of two lyric lines. We use the CMU pronunciation dictionary [1] to extract the phonetic transcription. For example, the phonetic transcription of "I love you" is [AY1, L, AH1, V, Y, UW1].

Part-of-speech similarity (sim_{pos}): To capture similarities in grammatical structure, we calculate a normalized Levenshtein edit distance between the part-of-speech (POS) sequences of two lines. We use the default POS tagger in the NLTK package [46].

Word vector similarity (sim_{w2v}): To capture the semantic similarity between two lyric lines, we simply average vectors of the words of each lyric line by using pre-trained word2vec [42] and compute their cosine similarity. This "bag of words" representation does not differentiate "dog bites person" from "person bites dog".

Context vector similarity (sim_{c2v}): To consider the word order, we vectorize the lyric lines using pre-trained context2vec [43], an extension of word2vec, which encodes a sequence of words by using Long Short-Term Memory (LSTM) networks [47]. We then compute their cosine similarity to obtain sim_{c2v}.

Word syllable count similarity (sim_{syW}): Since repeated phrases sometimes have the same number of syllables even if their words are different, we use a sequence of word syllable counts on each lyric line. For example, the word syllable counts of the two lyric lines *"Sometimes you lost yourself away"* and *"Everytime you just close your eyes"* [2] are $\{2, 1, 1, 2, 1\}$ and $\{2, 1, 1, 1, 1, 1\}$, respectively. When successive lyric lines have similar syllable count sequences, they are likely to correspond to the repetition of sections. We use dynamic time warping (DTW) [48] to calculate the similarity between syllable count sequences.

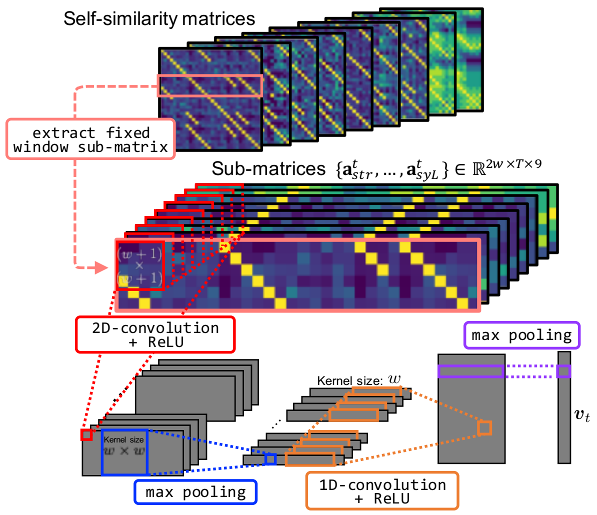

Figure 2. Convolutional neural network for SSMs.

Lyric Line syllable count similarity (sim_{syL}): We can also use the total syllable count of all words in each lyric line. For example, in all the chorus sections shown in Figure 1, the total syllable count of the first lyric line is 6 and that of the second line is 8. We calculate the similarity of such total syllable counts of each pair of lyric lines by using the following procedure. (1) We extract a window of four lyric lines $L_t = \{x_t, x_{t+1}, x_{t+2}, x_{t+3}\}$ and shift it over the entire lyrics of a song. (2) The similarity between the lyric lines x_t and $x_{t'}$ is calculated by DTW of L_t and $L_{t'}$.

We thus calculated nine SSMs $\mathbf{A}_m \in \mathbb{R}^{T \times T}$, where each cell is a sim_m explained above. Then, to calculate feature vectors from the above nine SSMs, we exploit a convolutional neural network (CNN) architecture to detect textual macro structures from various patterns in SSMs regardless of their locations and relative sizes in SSMs. Except for network parameters, this CNN architecture is the same as that of Fell et al. [18], as we share the same motivation: to extract translation, scaling and rotation invariant features from the input image (in our case, nine SSMs).

Figure 2 illustrates the CNN structure. After calculating the nine SSMs, we extract fixed-size elongated-rectangle sub-matrices centered on the target lyric line: $\mathbf{a}^t_m = \mathbf{A}_m[t - w + 1, ..., t + w; 1, ..., T] \in \mathbb{R}^{2w \times T}$, where w is a fixed window size. The input of the CNN is nine sub-matrices $\{\mathbf{a}^t_{str}, ..., \mathbf{a}^t_{syL}\} \in \mathbb{R}^{2w \times T \times 9}$, where the number of channels corresponds to the number of SSMs. The kernel size of the first 2D-convolutional layer is $(w+1) \times (w+1)$ so that each feature can capture a prospective chorus section. Each resulting tensor is downsampled by max-pooling with $w \times w$ kernel size. We then apply the 1D-convolutional layer with a kernel size of w and the last max-pooling layer downsamples the resulting vector to a scalar. In this network, all convolutional layers employ the ReLU function. We can perform the above procedure independently for each lyric line x_t and obtain the CNN-based feature vector $\boldsymbol{v}_t$.

3.2 Linguistic Features

Some expressions tend to appear in chorus sections. To quantify this tendency, we calculate the difference between

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

[2] This lyrics are taken from the RWC Music Database (RWC-MDB-P-2001 No.92) [44].

Tri-gram	$P_c - P_n$	Tri-gram	$P_n - P_c$
I'm	0.12%	there's	0.04%
don't	0.11%	I've	0.03%
oh oh oh	0.05%	's a	0.03%
I'll	0.05%	I'd	0.02%
we're	0.04%	but I'	0.02%
you're	0.04%	's not	0.01%
'll be	0.04%	what's	0.01%
I don'	0.04%	na na na	0.01%
Let's	0.03%	yeah yeah yeah	0.01%
you got ta	0.03%	've been	0.01%
I can'	0.03%	't take	0.01%
can't	0.03%	didn't	0.01%

Table 1. Frequent word tri-grams in chorus and non-chorus sections. An apostrophe is regarded as a word.

word tri-gram probabilities in the chorus and non-chorus sections. Table 1 shows the word tri-grams that frequently appear in both of the sections. Here, P_c and P_n denote word tri-gram probabilities in the chorus and non-chorus sections, respectively. As shown in this table, we found that phrases about the future (e.g., *"I'll"* and *"Let's"*) tend to appear in chorus sections more often than do phrases about the past (e.g., *"have been"* and *"didn't"*). To exploit such tendencies, we propose two linguistic features.

Average of word vectors ($ling_{ave}$): We use the average of word vectors of a given lyric line as a feature. The word vectors are obtained using pre-trained word2vec [42], skipping out-of-vocabulary words.

Vector representations for word sequences ($ling_{seq}$): To consider the word order that cannot be modeled by $ling_{ave}$, we use pre-trained context2vec [43] that enables vectorization of a lyric line by putting a sequence of word vectors into the LSTM.

We calculate the above linguistic feature vectors for each lyric line x_t and obtain their concatenated vector $\boldsymbol{u}_t$.

3.3 Neural Network-based Sequence Labeling Model

To solve the sequence labeling problem, we use the standard Bidirectional Long Short-Term Memory (Bi-LSTM) networks [49] to compute the conditional probability $P(Y_s|X_s)$. The neural network structure is illustrated in Figure 3.

The input to the Bi-LSTM layer at each time step t (lyric line) is a concatenation of two different types of feature vectors: (1) structural feature vectors $\boldsymbol{v}_t$ encoded from nine variations of SSMs in Section 3.1 and (2) linguistic feature vectors $\boldsymbol{v}_t$ encoded in Section 3.2. Formally, the conditional probability $P(Y_s|X_s)$ is calculated by using a $\mathrm{softmax}$ function:

$$P(Y_s|X_s) = \frac{\exp(\mathrm{Score}(X_s,Y_s))}{\sum_{Y'_s}\exp(\mathrm{Score}(X_s,Y'_s))}. \tag{1}$$

The $\mathrm{Score}()$ is defined as

$$\mathrm{Score}(X_s, Y_s) = \sum_t \mathrm{BN}(h_t[y_t]), \tag{2}$$

where $h_t[y_t]$ is the output of the Bi-LSTM for each time step t and $\mathrm{BN}()$ denotes batch normalization [50]. In the model training step, we use a binary cross-entropy loss.

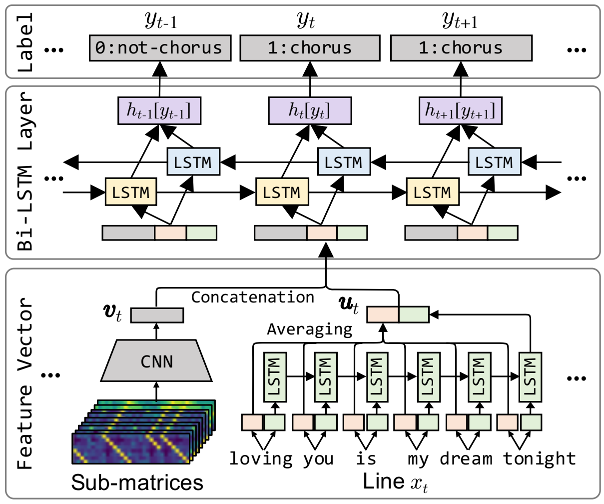

Figure 3. Neural-network-based sequence labeling model for chorus-section detection.

4. EXPERIMENT

Inspired by audio-based chorus-section detection [39], we evaluated the proposed method by using the F-measure (F) that is a harmonic mean of precision (P) and recall (R), $F = (2 \cdot R \cdot P)/(R + P)$, where

$$P = \frac{\text{\# of lyric lines in correctly detected chorus sections}}{\text{\# of lyric lines in detected chorus sections}}.$$

$$R = \frac{\text{\# of lyric lines in correctly detected chorus sections}}{\text{\# of lyric lines in correct (annotated) chorus sections}}.$$

We also used the pair-wise F-measure (p-F), normalized conditional entropy F-measure (n-F) and V-measure (V) that are provided by the Python module `mir_eval` and commonly used to evaluate computational music structure analysis [51].

4.1 Methods Compared

To confirm the effectiveness of our Bi-LSTM method based on the Bi-LSTM model that can learn dependencies between adjacent lyric lines, we compared its performance with that of with two baseline methods:

(1) Heuristic: We implemented the heuristic that "if lines at the end of the lyrics are repeated with small modifications, all those repeated lines are chorus sections" by the following procedure: (i) From the SSM that is the average of the nine SSMs, we extracted diagonals whose cells had values higher than a threshold λ, which was tuned on a development set to be $\lambda = 0.62$. (ii) From the extracted diagonals, we selected the shortest diagonal among diagonals placed at the bottom of the SSM (e.g., the diagonal starting at the cell $\mathbf{SSM}[29; 1]$ in Figure 1). (iii) Successive lines corresponding to the rows where the selected diagonal was located (e.g., lyric lines 29–32 in Figure 1) were assigned the label `chorus`. (iv) Other successive lines that were similar to the `chorus` lines (e.g., lyric lines 1–4 and 13–16 in Figure 1) were also assigned `chorus` labels.

(2) Multi-Layer Perceptron (MLP): Like the Bi-LSTM method, but with the Bi-LSTM model replaced by a standard MLP model. This method ignores transitions between adjacent lyric lines and predicts y_t from x_t only.

We chose the number of kernels for the first and second CNNs to be 200 and 400, respectively. We used $w = 3$ for the window size. In the MLP and Bi-LSTM methods, we chose the dimension of the hidden state to be 600. The word2vec [42] and context2vec [43] were pre-trained on lyrics and were not updated in the model training step of our method. The dimension of their output vectors was 300. We used AdamW for parameter optimization [52]. The initial learning rate was 0.001 with an exponential decay. We used a mini-batch size of 64. Training was run for 100 epochs, and the model used for testing was the one that achieved the best F-measure on the development set.

4.2 Dataset

To train our computational model that predicts whether the label of each lyric line is `chorus` or `not-chorus`, we needed a large amount of lyrics data with line-level chorus-section annotations like those illustrated in Figure 1. Since there was no dataset for this, we generated a large amount of such lyrics data by the following procedure:
(1) We prepared 100,772 pairs of musical audio signals and their corresponding manually time-aligned (temporally synchronized) lyrics [3]. To avoid unreliable lyrics, we confirmed that all lyrics had more than eight lines and less than 120 lines.
(2) We detected chorus sections of every song automatically by using its audio signals. In our experiments, we used the RefraiD method [39] to obtain the start and end times of each chorus section, but other methods could also be used.
(3) If the start time of a lyric line was within any chorus section detected in audio signals, that line was labeled `chorus`; otherwise, it was labeled `not-chorus`.
Of course, not all generated annotations were correct, but by using over 100,000 training data, the model could be robustly trained without being influenced by errors or outliers. The generated training data consisted of 9,313 English and 91,459 Japanese songs, and we called them EN_auto and JA_auto, respectively. [4]

Furthermore, we manually annotated three sets of lyrics data with more reliable line-level chorus-section annotations for three different purposes:
(a) For training comparison: We annotated 1,103 Japanese lyrics and called them JA_man [5]. By comparing the performance of the model trained on JA_auto with that of the model trained on JA_man, we could confirm that our generated data is reliable enough for training purposes.
(b) For tuning model parameters: We annotated the lyrics of 21 English and 79 Japanese songs from RWC-MDB-P-2001 and called them EN_RWC and JA_RWC, respectively. These were used to tune model parameters.
(c) For testing: We annotated the lyrics of 118 other English songs and 128 other Japanese songs and called them EN_test and JA_test, respectively [6]. These were used to test the chorus-section detection methods.

[3] In our experiments, English and Japanese lyrics text as well as the start time of every lyric line were provided by a lyrics distribution company. Automatic lyrics-to-audio synchronization [7–17] could also be used to estimate such start times.

[4] The main genres are Rock (33%), Pop (25%) and Alternative (12%) for EN_auto, and J-Pop (53%), Rock (20%) and Anime (9%) for JA_auto.

[5] To investigate the accuracy of the automatic annotation method we used for generating EN_auto and JA_auto, we applied the same method to the songs (audio signals and corresponding manually time-aligned lyrics) in JA_man. The accuracy of the generated annotations was $F = 68.0\%$, thus the automatic annotation method seems to work decently well.

	Training data / Testing data							
	EN_auto / EN_test				JA_auto / JA_test			
Method	F	$p\text{-}F$	$n\text{-}F$	V	F	$p\text{-}F$	$n\text{-}F$	V
Heuristic	57.8	73.8	43.0	35.8	57.1	73.2	43.6	36.3
MLP	74.2	76.8	47.7	43.0	80.6	82.8	62.6	59.1
Bi-LSTM	**78.1**	**77.7**	**50.8**	**47.3**	**83.4**	**83.5**	**64.9**	**61.4**

Table 2. Experimental result: Comparison of different methods (the unit is %).

	Training data / Testing data							
	EN_auto / EN_test				JA_auto / JA_test			
Feature	F	$p\text{-}F$	$n\text{-}F$	V	F	$p\text{-}F$	$n\text{-}F$	V
sim_{all}	77.9	76.1	48.6	45.5	81.2	82.7	63.6	59.6
$ling_{all}$	57.4	59.9	16.5	6.9	55.2	61.8	22.1	16.7
both	**78.1**	**77.7**	**50.8**	**47.3**	**83.4**	**83.5**	**64.9**	**61.4**

Table 3. Experimental result: Importance of using both structural and linguistic features.

4.3 Comparison of Different Methods

Table 2 summarizes the evaluated performances of Heuristic, MLP and the proposed Bi-LSTM. We found that MLP and Bi-LSTM performed better than Heuristic. This indicates that methods based on supervised learning are better than a rule-based method. We also found that Bi-LSTM was better than MLP and thus confirmed the importance of learning dependencies between adjacent lines. Since we concluded from these results that the proposed Bi-LSTM is the best for the chorus-section detection task, in the subsequent experiments reported here we used only Bi-LSTM.

4.4 Importance of Using Both Structural and Linguistic Features

To investigate the effectiveness of structural and linguistic features, we compared their use individually and in combination. Table 3 summarizes the results. We found that the model with only the structural features sim_{all} greatly outperformed the model with only the linguistic features $ling_{all}$. Using both kinds of features further improved the performance. This not only confirms the importance of using SSMs, as had been shown for the audio-based detection of chorus sections, but also confirms that the additional use of linguistic features is helpful for detecting chorus sections, which has not been shown before.

4.5 Reliability of Generated Annotations

As stated in Section 4.2, we used JA_man for the proposed training comparison. Table 4 clearly shows that the model trained using JA_auto, automatically generated data the amount of which can be large, outperformed the model trained using JA_man, manually annotated data, the amount

[6] The `chorus` and `not-chorus` labels were annotated only on the lyrics. No audio signal is available for these test data.

Training data	F	p-F	n-F	V
JA_auto (91,459 songs)	**83.4**	**83.5**	**64.9**	**61.4**
JA_man (1,103 songs)	80.3	77.3	53.3	50.4

Table 4. Experimental result: Reliability of automatically generated annotations.

Training data	Testing data	F	p-F	n-F	V
EN_auto (9,313 songs)	EN_test	77.9	76.1	48.6	45.5
JA_auto (91,459 songs)	EN_test	80.3	80.6	58.1	54.4
EJ_auto (100,772 songs)	EN_test	**81.0**	**82.3**	**60.7**	**57.4**

Table 5. Experimental result: Can the Japanese model detect English chorus sections?

of which is usually very limited because of the laborious manual effort its creation requires. The results also confirm that even if annotations generated automatically are not perfect they are reliable enough for training the model.

4.6 Training Data Size and Language Dependency

Tables 2 and 3 also show that the performances for English lyrics were worse than those for Japanese lyrics. Since the amount of Japanese training data was about 10 times than that of English training data, we think that the amount of training data greatly affects the performance of the proposed model. We are thus interested in answering the question "Can a model trained on a large amount of Japanese data detect English chorus sections?" In fact, although linguistic features are language dependent and the process of computing SSMs is also language dependent, structural features based on the resulting SSMs can be language independent because our SSMs simply represent patterns of repeating lyric lines, which could be universal in music.

As shown in the upper half of Table 5, which shows results obtained without using linguistic features, we found that the structural-feature-based model trained on Japanese data JA_auto succeeded in detecting English chorus sections in EN_test and its performance was better than that of the model trained on the smaller dataset EN_auto. This result indicates that the SSM-based model trained on a large amount of data can detect chorus sections regardless of the language of the test set. Moreover, this result is further evidence that Japanese and English SSMs (i.e., patterns of repeating lyric lines) have similar structures.

Obviously, the above result raises another question: "Can a model trained on both EN_auto and JA_auto perform better than one trained on only EN_auto or JA_auto?" To answer this question, we created training data EJ_auto by including both EN_auto and JA_auto and constructed yet another structural-feature-based model with EJ_auto. As shown in the lower half of Table 5, we found that the model trained on both languages performed better than the model trained on only one.

These results confirm that chorus sections can be detected by a model trained on data in another language, that patterns of repeating lyric lines are language-independent and that mixing different language data allows the model to learn the general structure of chorus sections and thereby perform better. This could have an impact on low-resource languages because large-scale training data can be created by mixing other available language resources.

5. RELATED WORK

Previous work in the MIR community has addressed musical structure analysis and chorus-section detection based on repeated patterns in musical audio signals [23–41]. Studies in the chorus-section detection for audio signals typically used SSMs to capture repeated structures, and we share this motivation. Our approach differs from those audio-based approaches in that it exploits multiple lyrics-based SSMs and linguistic features within chorus sections.

On the other hand, recent work in the NLP community has tackled lyrics segmentation and summarization tasks by exploiting SSMs. Fell et al. and Watanabe et al. proposed a neural network model and logistic regression model for segmenting paragraphs (sections) without labeling them by using SSMs as features [18, 19]. Those tasks, however, are essentially different from detecting all chorus sections that are the most representative sections in lyrics text. Addressing a task similar to chorus-section detection, Fell et al. [53] proposed a method of summarizing lyrics by combining general document summarization methods with audio thumbnailing methods. They focus on extracting individual informative lines as a summary from lyrics text, not redundant repeated lines. On the other hand, the focus of our paper is to detect chorus sections whose successive lines are often repeated in lyrics text.

6. CONCLUSION

This paper has addressed the novel task of detecting chorus sections in English and Japanese lyrics. We proposed a neural-network-based sequence labeling model that learns structural (i.e., phrase-repetition) and linguistic features to detect lyric lines of chorus sections. We also generated over 100,000 training data with chorus-section annotations. No previous work has ever conducted chorus-section detection for text-only lyrics with this much data.

The contributions of this study are summarized as follows: (1) We designed a variety of features to capture structural and linguistic properties of chorus sections. (2) We proposed a sequence labeling model that can detect chorus sections in lyrics. (3) We showed how to generate a large training dataset of lyrics with chorus-section annotations. (4) We demonstrated that our Bi-LSTM-based method outperforms alternative baseline methods. (5) We thoroughly investigated this detection task and the nature of chorus sections of lyrics from different perspectives such as the importance of features, the amount of training data, and language dependency.

We plan to extend our method to detect other sections, such as verse and bridge sections. Future work will also develop MIR applications using our method, such as those discussed in Section 1.

7. ACKNOWLEDGMENTS

The authors appreciate SyncPower Corporation for providing lyrics data. This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

8. REFERENCES

[1] K. Watanabe and M. Goto, "Query-by-Blending: A music exploration system blending latent vector representations of lyric word, song audio, and artist," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019, pp. 144–151.

[2] K. Tsukuda, K. Ishida, and M. Goto, "Lyric Jumper: A lyrics-based music exploratory web service by modeling lyrics generative process," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 544–551.

[3] S. Sasaki, K. Yoshii, T. Nakano, M. Goto, and S. Morishima, "LyricsRadar: A lyrics retrieval system based on latent topics of lyrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 585–590.

[4] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017, pp. 694–701.

[5] R. Mayer and A. Rauber, "Music genre classification by ensembles of audio and lyrics features," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 675–680.

[6] R. Mayer, R. Neumayer, and A. Rauber, "Rhyme and style features for musical genre classification by song lyrics," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008, pp. 337–342.

[7] B. Sharma, C. Gupta, H. Li, and Y. Wang, "Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 396–400.

[8] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," in *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 181–185.

[9] C. Gupta, E. Yilmaz, and H. Li, "Acoustic modeling for automatic lyrics-to-audio alignment," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019)*, 2019, pp. 2040–2044.

[10] C. Gupta, R. Tong, H. Li, and Y. Wang, "Semi-supervised lyrics and solo-singing alignment," in *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, 2018, pp. 600–607.

[11] S. Chang and K. Lee, "Lyrics-to-audio alignment by unsupervised discovery of repetitive patterns in vowel acoustics," *IEEE Access*, vol. 5, pp. 16 635–16 648, 2017.

[12] S. W. Lee and J. Scott, "Word level lyrics-audio synchronization using separated vocals," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2017)*, 2017, pp. 646–650.

[13] Y. Chien, H. Wang, and S. Jeng, "Alignment of lyrics with accompanied singing audio based on acoustic-phonetic vowel likelihood modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 1998–2008, 2016.

[14] G. B. Dzhambazov and X. Serra, "Modeling of phoneme durations for alignment between polyphonic audio and lyrics," in *Proceedings of the 12th Sound and Music Computing Conference (SMC 2015)*, 2015.

[15] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 200–210, 2012.

[16] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "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.

[17] M. Kan, Y. Wang, D. Iskandar, T. L. Nwe, and A. Shenoy, "LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals," *IEEE Trans. Speech Audio Process.*, vol. 16, no. 2, pp. 338–349, 2008.

[18] M. Fell, Y. Nechaev, E. Cabrio, and F. Gandon, "Lyrics segmentation: Textual macrostructure detection using convolutions," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, 2018, pp. 2044–2054.

[19] K. Watanabe, Y. Matsubayashi, N. Orita, N. Okazaki, K. Inui, S. Fukayama, T. Nakano, J. B. L. Smith, and M. Goto, "Modeling discourse segments in lyrics using repeated patterns," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, 2016, pp. 1959–1969.

[20] A. Baratè, L. A. Ludovico, and E. Santucci, "A semantics-driven approach to lyrics segmentation," in

Proceedings of the 8th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP 2013), 2013, pp. 73–79.

[21] J. P. G. Mahedero, A. Martinez, P. Cano, M. Koppenberger, and F. Gouyon, "Natural language processing of lyrics," in *Proceedings of the 13th ACM International Conference on Multimedia (ACM Multimedia)*, 2005, pp. 475–478.

[22] K. Watanabe, Y. Matsubayashi, K. Inui, S. Fukayama, T. Nakano, and M. Goto, "Modeling storylines in lyrics," *IEICE Transactions on Information and Systems*, vol. 101-D, no. 4, pp. 1167–1179, 2018.

[23] G. Shibata, R. Nishikimi, E. Nakamura, and K. Yoshii, "Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularity-aware hierarchical hidden semi-markov model," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019, pp. 268–275.

[24] A. Maezawa, "Music boundary detection based on a hybrid deep model of novelty, homogeneity, repetition and duration," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2019)*, 2019, pp. 206–210.

[25] G. Sargent, F. Bimbot, and E. Vincent, "Estimating the structural segmentation of popular music pieces under regularity constraints," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 344–358, 2017.

[26] T. Cheng, J. B. L. Smith, and M. Goto, "Music structure boundary detection and labelling by a deconvolution of path-enhanced self-similarity matrix," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2018)*, 2018, pp. 106–110.

[27] J. B. L. Smith and M. Goto, "Using priors to improve estimates of music structure," in *Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016)*, 2016, pp. 554–560.

[28] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*, 2015, pp. 531–537.

[29] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 405–410.

[30] G. Peeters and V. Bisot, "Improving music structure segmentation using lag-priors," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 337–342.

[31] H. Grohganz, M. Clausen, N. Jiang, and M. Müller, "Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013, pp. 209–214.

[32] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2013)*, 2013, pp. 236–240.

[33] F. Kaiser and G. Peeters, "A simple fusion method of state and sequence segmentation for music structure discovery," in *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013, pp. 257–262.

[34] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised detection of music boundaries by time series structure features," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 2012.

[35] M. Müller, P. Grosche, and N. Jiang, "A segment-based fitness measure for capturing repetitive structures of music recordings," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 615–620.

[36] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010, pp. 625–636.

[37] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, 2009.

[38] M. Müller and S. Ewert, "Joint structure analysis with applications to music annotation and synchronization," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008, pp. 389–394.

[39] M. 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.

[40] M. Cooper and J. Foote, "Summarizing popular music via structural similarity analysis," in *Proceedings of the 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2003)*, 2003, pp. 127–130.

[41] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of the 2000 IEEE International Conference on Multimedia and Expo (IEEE ICME 2000)*, 2000, p. 452.

[42] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NeurIPS 2013)*, 2013, pp. 3111–3119.

[43] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional LSTM," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 51–61.

[44] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, classical and jazz music databases," in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, vol. 2, 2002, pp. 287–288.

[45] Y. Li and B. Liu, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[46] S. Bird, "NLTK: the natural language toolkit," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, 2006.

[47] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[48] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proceedings of Workshop on Knowledge Discovery in Databases*, 1994, pp. 359–370.

[49] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP 2013)*, 2013, pp. 6645–6649.

[50] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, vol. 37, 2015, pp. 448–456.

[51] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common MIR metrics," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014, pp. 367–372.

[52] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, 2019.

[53] M. Fell, E. Cabrio, F. Gandon, and A. Giboin, "Song lyrics summarization inspired by audio thumbnailing," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 328–337.