A Melody-conditioned Lyrics Language Model

Kento Watanabe[1], **Yuichiroh Matsubayashi**[1],
Satoru Fukayama[2], **Masataka Goto**[2], **Kentaro Inui**[1,3], **Tomoyasu Nakano**[2]
[1]Graduate School of Information Sciences, Tohoku University,
[2]National Institute of Advanced Industrial Science and Technology (AIST),
[3]RIKEN Center for Advanced Intelligence Project
{kento.w, y-matsu, inui}@ecei.tohoku.ac.jp,
{s.fukayama, m.goto, t.nakano}@aist.go.jp

Abstract

This paper presents a novel, data-driven language model that produces entire lyrics for a given input melody. Previously proposed models for lyrics generation suffer from the inability of capturing the relationship between lyrics and melody partly due to the unavailability of lyrics-melody aligned data. In this study, we first propose a new practical method for creating a large collection of lyrics-melody aligned data and then create a collection of 1,000 lyrics-melody pairs augmented with precise syllable-note alignments and word/sentence/paragraph boundaries. We then provide a quantitative analysis of the correlation between word/sentence/paragraph boundaries in lyrics and melodies. We then propose an RNN-based lyrics language model conditioned on a featurized melody. Experimental results show that the proposed model generates fluent lyrics while maintaining the compatibility between boundaries of lyrics and melody structures.

1 Introduction

Writing lyrics for a given melody is a challenging task. Unlike prose text, writing lyrics requires both knowledge and consideration of music-specific properties such as the structure of melody, rhythms, etc. (Austin et al., 2010; Ueda, 2010). A simple example is the correlation between word boundaries in lyrics and the rests in a melody. As shown in Figure 1, a single word spanning beyond a long melody rest can sound unnatural. When writing lyrics, a lyricist must consider such constraints in content and lexical selection, which can impose extra cognitive loads.

This consideration when writing lyrics has motivated a wide-range of studies for the task of computer-assisted lyrics writing (Barbieri et al., 2012; Abe and Ito, 2012; Potash et al., 2015; Watanabe et al., 2017). Such studies aim to model the language in lyrics and to design a computer system for assisting lyricists in writing. They propose to constrain their models to generate only lyrics that satisfy given conditions on syllable counts, rhyme positions, etc. However, such constraints are assumed to be manually provided by a human user, which requires the user to interpret a source melody and transform their interpretation to a set of constraints. To assist users with transforming a melody to constraints, a language model that automatically captures the relationship between lyrics and melody is required.

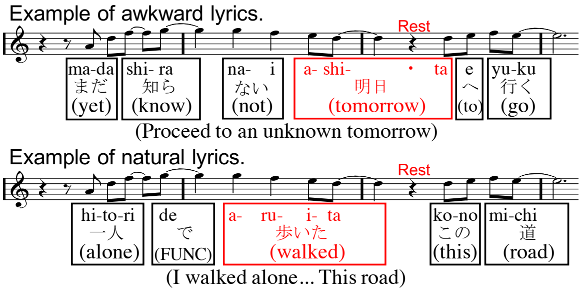

Figure 1: Examples of awkward and natural lyrics. FUNC indicates a function word. The song is from the RWC Music Database (RWC-MDB-P-2001 No.20) (Goto et al., 2002).

Some studies (Oliveira et al., 2007; Oliveira, 2015; Nichols et al., 2009) have quantitatively analyzed the correlations between melody and phonological aspects of lyrics (e.g., the relationship between a beat and a syllable stress). However, these studies do not address the relationship between melody and the *discourse structure* of lyrics. Lyrics are not just a sequence of syllables but a meaningful sequence of words. Therefore, it is desirable that the sentence/paragraph boundaries are determined based on both melody rests and context words.

Considering such line/paragraph structure of lyrics, we present a novel language model that gen-

Proceedings of NAACL-HLT 2018, pages 163–172
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

erates lyrics whose word, sentence, and paragraph boundaries are appropriate for a given melody, without manually transforming the melody to syllable constraints. This direction of research has received less attention because it requires a large dataset consisting of aligned pairs of melody and segment boundaries of lyrics which has yet to exist.

To address this issue, we leverage a publicly-available collection of digital music scores and create a dataset of digital music scores each of which specifics a melody score augmented with syllable information for each melody note. We collected 1,000 Japanese songs from an online forum where many amateur music composers upload their music scores. We then automatically aligned each music score with the raw text data of the corresponding lyrics in order to augment it with the word, sentence, and paragraph boundaries.

The availability of such aligned, parallel data opens a new area of research where one can conduct a broad range of data-oriented research for investigating and modeling correlations between melodies and discourse structure of lyrics. In this paper, with our melody-lyrics aligned songs, we investigate the phenomena that (i) words, sentences, and paragraphs rarely span beyond a long melody rest and (ii) the boundaries of larger components (i.e., paragraphs) tend to coincide more with longer rests. To the best of our knowledge, there is no previous work that provides any quantitative analysis of this phenomenon with this size of data (see Section 7).

Following this analysis, we build a novel, data-driven language model that generates fluent lyrics whose sentence and paragraph boundaries fit an input melody. We extend a Recurrent Neural Network Language Model (RNNLM) (Mikolov et al., 2010) so that its output can be conditioned on a featurized melody. Both our quantitative and qualitative evaluations show that our model captures the consistency between melody and boundaries of lyrics while maintaining word fluency.

2 Melody-lyric alignment data

Our goal is to create a melody-conditioned language model that captures the correlations between melody patterns and discourse segments of lyrics. The data we need for this purpose is a collection of melody-lyrics pairs where the melody and lyrics are aligned at the level of not only note-syllable alignment but also discourse components

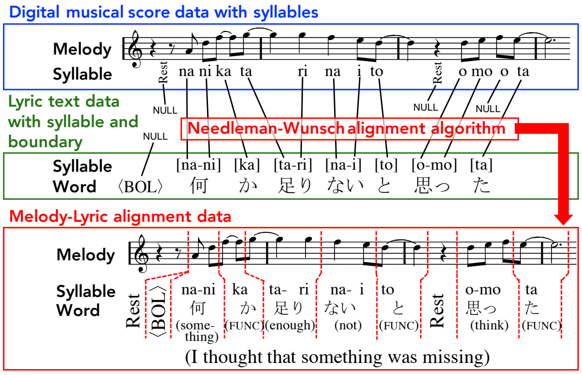
Digital musical score data with syllables
Melody
Syllable
Rest
na ni ka ta ri na i to
Rest
o mo o ta
NULL
NULL
NULL
NULL
Lyric text data with syllable and boundary
Needleman-Wunsch alignment algorithm
Syllable
Word
[na-ni] [ka] [ta-ri] [na-i] [to] [o-mo] [ta]
〈BOL〉 何 か 足り ない と 思っ た
Melody-Lyric alignment data
Melody
Syllable
Word
Rest
〈BOL〉
na-ni 何 (some-thing)
ka か (FUNC)
ta- ri 足り (enough)
na- i ない (not)
to と (FUNC)
Rest
o-mo 思っ (think)
ta た (FUNC)
(I thought that something was missing)

Figure 2: Melody-lyrics alignment using the Needleman Wunsch algorithm. BOL denotes a line boundary.

(i.e., word/sentence/paragraph boundaries) of a lyric, as illustrated in the bottom of Figure 2. We create such a dataset by automatically combining two types of data available from online forum sites: digital music score data (the top of Figure 2) and raw lyrics data (the middle).

A digital music score specifies a melody score augmented with syllable information for each melody note (see the top of Figure 2). Score data augmented in this way is sufficient for analyzing the relationship between the phonological aspects of lyrics and melody, but it is insufficient for our goal since the structural information of the lyrics is not included. We thus augment score data further with boundaries of sentences, and paragraphs, where we assume that sentences and paragraphs of lyrics are approximately captured by *lines* and *blocks*,[1] respectively, of the lyrics in the raw text.

The integration of music scores and raw lyrics is achieved by (1) applying a morphological analyzer[2] to raw lyrics for word segmentation and Chinese character pronunciation prediction and (2) aligning music score with raw lyrics at the syllable level as illustrated in Figure 2. For this alignment, we employ the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). This alignment process is reasonably accurate because it fails in principle only when the morphological analysis fails in Chinese character pronunciation prediction, which occurs for only less than 1% of the words in the data set.

With this procedure, we obtained 54,181 Japanese raw lyrics and 1,000 digital musical

[1] Blocks are assumed to be segmented by empty lines.

[2] To extract word boundaries and syllable information for Japanese lyrics, we apply MeCab parser (Kudo et al., 2004).

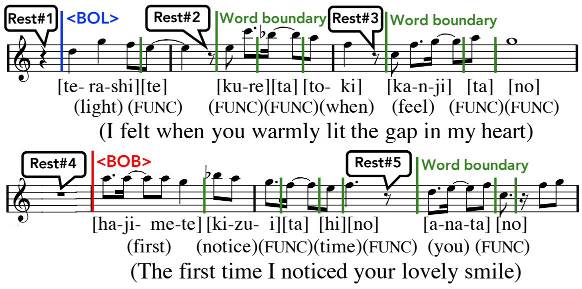
Rest#1
<BOL>
Rest#2
Word boundary
Rest#3
Word boundary
[te- ra-shi][te] [ku-re][ta] [to- ki] [ka-n-ji] [ta] [no]
(light) (FUNC) (FUNC)(FUNC)(when) (feel) (FUNC)(FUNC)
(I felt when you warmly lit the gap in my heart)
Rest#4
<BOB>
Rest#5
Word boundary
[ha-ji- me-te] [ki-zu- i][ta] [hi][no] [a-na-ta] [no]
(first) (notice)(FUNC)(time)(FUNC) (you) (FUNC)
(The first time I noticed your lovely smile)

Figure 3: Example boundaries appearing immediately after a rest. BOB indicates a block boundary.

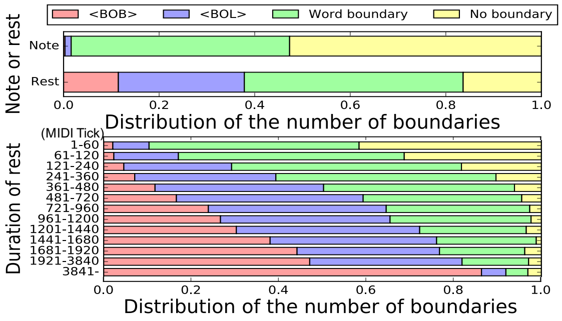

Figure 4: Distribution of the number of boundaries in the melody-lyrics alignment data.

scores from online forum sites[3]; we thus created 1,000 melody-lyrics pairs. We refer to these 1,000 melody-lyrics pairs as a *melody-lyrics alignment* data[4] and refer to the remaining 53,181 lyrics without melody as a *raw lyrics* data.

We randomly split the 1,000 melody-lyrics alignments into two sets: 90% for analyzing/training and the remaining 10% for testing. From those, we use 20,000 of the most frequent words whose syllable counts are equal to or less than 10, and converted others to a special symbol ⟨unknown⟩. All of the digital music score data we collected were distributed in the UST format, a common file format designed specifically for recently emerging computer vocal synthesizers. While we focus on Japanese music in this study, our method for data creation is general enough to be applied to other language formats such as MusicXML and ABC, because transferring such data formats to UST is straightforward.

3 Correlations between melody and lyric

In this section, we examine two phenomena related to boundaries of lyrics: (1) the positions of lyrics segment boundaries are biased to melody rest positions, and (2) the probability of boundary occurrence depends on the duration of a rest, i.e., a shorter rest tends to be a word boundary and a longer rest tends to be a block boundary, as shown in Figure 3. All analyses were performed on the training split of the melody-lyrics alignment data, which is described in Section 2.

For the first phenomenon, we first calculated the distribution of boundary appearances at the positions of melody notes and rests. Here, by the *boundary of a line* (or block), we refer to the position of the beginning of the line (or block).[5] In Figure 3, we say, for example, that the boundary of the first block beginning "*te-ra-shi te*" coincides with Rest#1. The result, shown at the top of Figure 4, indicates that line and block boundaries are strongly biased to rest positions and are far less likely to appear at note positions. Words, lines, and blocks rarely span beyond a long melody rest.

The bottom of Figure 4 shows the detailed distributions of boundary occurrences for different durations of melody rests, where durations of 480 and 1920 correspond to a quarter rest and a whole rest, respectively. The results exhibit a clear, strong tendency that the boundaries of larger segments tend to coincide more with longer rests. To the best of our knowledge, this is the first study that has ever provided such strong empirical evidence for the phenomena related to the correlations between lyrics segments and melody rests. It is also important to note that the choice of segment boundaries looks like a probabilistic process (i.e., there is a long rest without a block boundary). This observation suggests the difficulty of describing the correlations of lyrics and melody in a rule-based fashion and motivates our probabilistic approach as we present in the next section.

4 Melody-conditioned language model

Our goal is to build a language model that generates fluent lyrics whose discourse segment fit a given melody in the sense that generated segment boundaries follow the distribution observed in Section 3. We propose to pursue this goal by conditioning a

[3] For selecting the 1,000 songs, we chose only frequently downloaded or highly popular songs to ensure the quality of the resulting dataset.

[4] We publicly release all source URLs of the 1,000 songs (`https://github.com/KentoW/melody-lyrics`).

[5] The beginning of a line/block and the end of a line/block are equivalent since there is no melody between the end and beginning of a line/block.

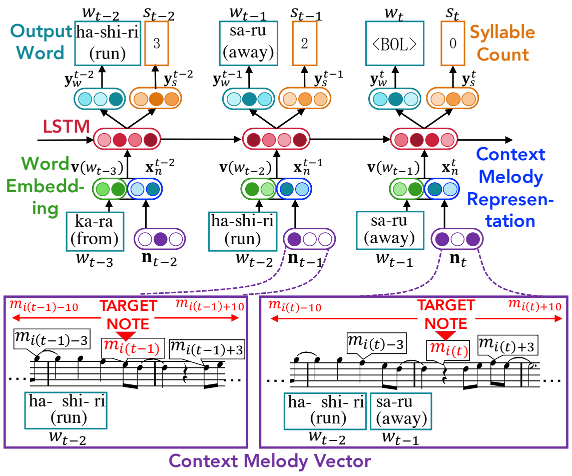
Output Word
Syllable Count
LSTM
Word Embedd-ing
Context Melody Represen-tation
w_{t-2} s_{t-2}
ha-shi-ri (run)
3
w_{t-1} s_{t-1}
sa-ru (away)
2
w_t s_t
⟨BOL⟩
0
$\mathbf{y}_w^{t-2}$ $\mathbf{y}_s^{t-2}$
$\mathbf{y}_w^{t-1}$ $\mathbf{y}_s^{t-1}$
$\mathbf{y}_w^{t}$ $\mathbf{y}_s^{t}$
$\mathbf{v}(w_{t-3})$ $\mathbf{x}_n^{t-2}$
$\mathbf{v}(w_{t-2})$ $\mathbf{x}_n^{t-1}$
$\mathbf{v}(w_{t-1})$ $\mathbf{x}_n^{t}$
ka-ra (from)
w_{t-3} $\mathbf{n}_{t-2}$
ha-shi-ri (run)
w_{t-2} $\mathbf{n}_{t-1}$
sa-ru (away)
w_{t-1} $\mathbf{n}_t$
$m_{i(t-1)-10}$ TARGET NOTE $m_{i(t-1)+10}$
$m_{i(t-1)-3}$ $m_{i(t-1)}$ $m_{i(t-1)+3}$
ha- shi- ri (run)
w_{t-2}
$m_{i(t)-10}$ TARGET NOTE $m_{i(t)+10}$
$m_{i(t)-3}$ $m_{i(t)}$ $m_{i(t)+3}$
ha- shi- ri (run)
sa-ru (away)
w_{t-2} w_{t-1}
Context Melody Vector

Figure 5: Melody-conditioned RNNLM.

standard RNNLM with a featurized input melody. We call this model a *Melody-conditioned RNNLM*.

The network structure of the model is illustrated in Figure 5. Formally, we are given a melody $\mathbf{m} = m_1,...,m_i,...,m_I$ that is a sequence of notes and rests, where m includes a pitch and a duration information. Our model generates lyrics $\mathbf{w} = w_1,...,w_t,...,w_T$ that is a sequence of words and segment boundary symbols: ⟨BOL⟩ and ⟨BOB⟩, special symbols denoting a line and a block boundary, respectively. For each time step t, the model outputs a single word or boundary symbol taking a pair of the previously generated word w_{t-1} and the musical feature vector $\mathbf{n}_t$ for the current word position which includes context window-based features that we describe in the following section. In this model, we assume that the syllables of the generated words and the notes in the input melody have a one-to-one correspondence. Therefore, the position of the incoming note/rest for a word position t (referred to as a target note for t) is uniquely determined by the syllable counts of the previously generated words.[6] The target note for t is denoted as $m_{i(t)}$ by defining a function $i(\cdot)$ which maps time step t to the index of the next note in t.

Here, the challenging issue with this model is training. Generally, language models require a large amount of text data to learn well. Moreover, this is also the case for learning correlation between rest positions and *syllable counts*. As shown in Figure 4, most words are supposed to not overlap a long rest. This means, for example, that when the incoming melody sequence for a next word position is *note, note, (long) rest, note, note*, as the sequence following to $m_{i(t-1)}$ in Figure 5, it is desirable to select a word whose syllable count is two or less so that the generated word does not overlap the long rest. If there is sufficient data available, this tendency may be learned directly from the correlation between rests and words without explicitly considering the syllable count of a word. However, our melody-lyrics alignments for 1,000 songs are insufficient for this purpose.

We take two approaches to address this data sparsity problem. First, we propose two training strategies that increase the number of training examples using raw lyrics that can be obtained in greater quantities. Second, we construct a model that predicts the number of syllables in each word, as well as words themselves, to explicitly supervise the correspondence between rest positions and syllable counts.

In the following sections, we first describe the details of the proposed model and then present the training strategies used to obtain better models with our melody-lyrics alignment data.

4.1 Model construction

The proposed model is based on a standard RNNLM (Mikolov et al., 2010):

$$P(\mathbf{w}) = \textstyle\prod_{t=1}^{T} P(w_t|w_0,...,w_{t-1}), \quad (1)$$

where context words are encoded using LSTM (Hochreiter and Schmidhuber, 1997) and the probabilities over words are calculated by a softmax function. $w_0 = \langle \mathrm{B} \rangle$ is a symbol denoting the beginning of lyrics. We extend this model such that each output is conditioned by the context melody vectors $\mathbf{n}_1,...,\mathbf{n}_t$, as well as previous words:

$$P(\mathbf{w}|\mathbf{m}) = \textstyle\prod_{t=1}^{T} P(w_t|w_0,...,w_{t-1},\mathbf{n}_1,...,\mathbf{n}_t). \quad (2)$$

The model simultaneously predicts the syllable counts of words by sharing the parameters of LSTM with the above word prediction model in order to learn the correspondence between the melody segments and syllable counts:

$$P(\mathbf{s}|\mathbf{m}) = \textstyle\prod_{t=1}^{T} P(s_t|w_0,...,w_{t-1},\mathbf{n}_1,...,\mathbf{n}_t), \quad (3)$$

where $\mathbf{s} = s_1,...,s_T$ is a sequence of syllable counts, which corresponds to $\mathbf{w}$.

[6] Note that our melody-lyrics alignment data used in training does not make this assumption, but we can still uniquely identify the positions of target notes based on the obtained melody-word alignment.

For each time step t, the model outputs a word distribution $\mathbf{y}_w^t \in \mathbb{R}^V$ and a distribution of syllable count $\mathbf{y}_s^t \in \mathbb{R}^S$ using a softmax function:

$$\mathbf{y}_w^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_w \mathbf{z}_t)), \tag{4}$$
$$\mathbf{y}_s^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_s \mathbf{z}_t)), \tag{5}$$

where $\mathbf{z}_t$ is the output of the LSTM for each time step. V is the vocabulary size and S is the syllable count threshold.[7] $\mathbf{W}_w$ and $\mathbf{W}_s$ are weight matrices. BN denotes batch normalization (Ioffe and Szegedy, 2015).

The input to the LSTM in each time step t is a concatenation of the embedding vector of the previous word $\mathbf{v}(w_{t-1})$ and the context melody representation $\mathbf{x}_n^t$, which is a nonlinear transformation of the context melody vector $\mathbf{n}_t$:

$$\mathbf{x}^t = [\mathbf{v}(w_{t-1}), \mathbf{x}_n^t], \tag{6}$$
$$\mathbf{x}_n^t = \mathrm{ReLU}(\mathbf{W}_n \mathbf{n}_t + \mathbf{b}_n), \tag{7}$$

where $\mathbf{W}_n$ is a weight matrix and $\mathbf{b}_n$ is a bias.

To generate lyrics, the model searches for the word sequence with the greatest probability (Eq. 2) using beam search. The model stops generating lyrics when the syllable count of the lyrics reaches the number of notes in the input melody.

Note that our model is not specific to the language of lyrics. The model only requires the sequences of melody, words, and syllable counts and does not use any language-specific features.

4.2 Context melody vector

In Section 3, we indicated that the positions of rests and their durations are important factors for modeling boundaries of lyrics. Thus, we collect a sequence of notes and rests around the current word position (i.e., time step t) and encode their information into context melody vector $\mathbf{n}_t$ (see the bottom of Figure 5).

The context melody vector $\mathbf{n}_t$ is a binary feature vector that includes a musical notation type (i.e., note or rest), a duration[8], and a pitch for each note/rest in the context window. We collect notes and rests around the target note $m_{i(t)}$ for the current word position t with a window size of 10 (i.e., $m_{i(t)-10}, ..., m_{i(t)}, ..., m_{i(t)+10}$).

For pitch information, we use a gap (pitch interval) between a target note $m_{i(t)}$ and its previous note $m_{i(t-1)}$. Here, the pitch is represented by a MIDI note number in the range 0 to 127. For example, the target and its previous notes are 68 and 65, respectively, and the gap is $+3$.

Algorithm 1 Pseudo melody generation

```
for each syllable in the input-lyrics do
    $b \leftarrow$ get boundary type next to the syllable
    sample note pitch $p \sim P(p_i | p_{i-2}, p_{i-1})$
    sample note duration $d_{\text{note}} \sim P(d_{\text{note}} | b)$
    assign note with $(p, d_{\text{note}})$ to the syllable
    sample binary variable $r \sim P(r | b)$
    if $r = 1$ then
        insert rest with duration $d_{\text{rest}} \sim P(d_{\text{rest}} | b)$
    end if
end for
```

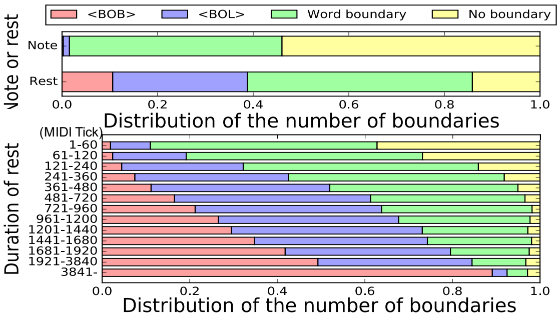

Figure 6: Distribution of the number of boundaries in pseudo-data.

4.3 Training strategies

Pretraining The size of our melody-lyrics alignment data is limited. However, we can obtain a large amount of raw lyrics. We, therefore, pretrain the model with 53,181 raw lyrics and then fine-tune it with the melody-lyrics alignment data. In pretraining, all context melody vectors $\mathbf{n}_t$ are zero vectors. We refer to these pretrained and fine-tuned models as *Lyrics-only* and *Fine-tuned* models, respectively.

Learning with pseudo-melody We propose a method to increase the melody-lyrics alignment data by attaching *pseudo melodies* to the obtained 53,181 raw lyrics. We refer to the model that uses this data as the *Pseudo-melody* model.

Algorithm 1 shows the details of pseudo-melody generation. For each syllable in the lyrics, we first assign a note to the syllable by sampling the probability distributions. The pitch of each note is generated based on the trigram probability. Then, we determine whether to generate a rest next to it. Since we established the correlations between rests and boundaries of lyrics in Section 3, the probability for a rest and its duration is conditioned by a boundary

[7] The syllable counts of the ⟨BOL⟩ and ⟨BOB⟩ are zero.

[8] We rounded each duration to one of the values 60, 120, 240, 360, 480, 720, 960, 1200, 1440, 1680, 1920, and 3840 and use one-hot encoding for each rounded duration.

type next to the target syllable. All probabilities are calculated using the training split of the melody-lyrics alignment data.

Figure 6 shows the distributions of the number of boundaries in the pseudo data. The distributions closely resemble those of gold data in Figure 4.

5 Quantitative evaluation

We evaluate the proposed Melody-conditioned RNNLMs quantitatively based on two evaluation metrics: (1) a test set perplexity for measuring the fluency; (2) a line/block boundary replication task for measuring the consistency between the melody and boundaries in the generated lyrics.

5.1 Experimental setup

In our model, we chose the dimensions of the word embedding vectors and context melody representation vectors to 512 and 256, respectively, and the dimension of the LSTM hidden state was 768. We used a categorical cross-entropy loss for outputs $\mathbf{y}_w^t$ and $\mathbf{y}_s^t$, Adam (Kingma and Ba, 2014) with an initial learning rate of 0.001 for parameter optimization, and a mini-batch size of 32. We applied an early-stopping strategy with a maximum epoch number of 100, and training was terminated after five epochs of unimproved loss on the validation set. For lyrics generation, we used a beam search with a width of 10. An example of the generated lyrics is shown in the supplemental material.

5.2 Evaluation metrics

Perplexity Test-set perplexity (PPL) is a standard evaluation measure for language models. PPL measures the predictability of wording in original lyrics, where a lower PPL value indicates that the model can generate fluent lyrics. We used PPL and its variant PPL-W, which excludes line/block boundaries, to investigate the predictability of words.

Accuracy of boundary replication Under the assumption that the line and block boundaries of the original lyrics are placed at appropriate positions in the melody, we evaluated consistency between the melody and boundaries in the generated lyrics by measuring the reproducibility of the boundaries in the original lyrics. Here the metric we used was F_1-measure of the boundary positions. We also asked a person to place line and block boundaries at plausible positions for randomly selected 10 input melodies that the evaluator has never heard. This person is not a professional musician but an experienced performer educated on musicology. The bottom part of Table 1 represents the human performance.

	Perplexity		F_1-measure		
Model	PPL	PPL-W	BOB	BOL	UB
Lyrics-only	138.0	225.0	0.121	0.061	0.106
Full-data	135.9	222.1	0.122	0.063	0.108
Alignment-only	173.3	314.8	0.298	0.287	0.477
Heuristic	175.8	284.7	**0.373**	0.239	0.402
Fine-tuned	152.2	275.5	0.260	**0.302**	**0.479**
Pseudo-melody	**115.7**	**197.5**	0.318	0.241	0.406
(w/o $\mathbf{y}_s$)					
Fine-tuned	155.1	278.1	0.318	0.241	0.366
Pseudo-melody	118.0	201.5	0.312	0.250	0.406
Human	-	-	0.717	0.671	0.751

Table 1: Results of the quantitative evaluation. "UB" denotes the score for unlabeled matching of line/block boundaries. "w/o $\mathbf{y}_s$" denotes the exclusion of the syllable-count output layer.

5.3 Effect of Melody-conditioned RNNLM

To investigate the effect of our language models, we compared the following six models. The first one is (1) a *Lyrics-only* model, a standard RNNLM trained with 54,081 song lyrics without melody information. The second and third ones are baseline Melody-conditioned RNNLMs where the proposed training strategies are not applied: (2) a *Full-data* model trained with mixed data (54,081 song lyrics and 900 melody-lyrics alignments of those), and (3) an *Alignment-only* model trained with only 900 melody-lyrics alignment data. The fourth one is a strong baseline to evaluate the performance of the proposed approaches: (4) a *Heuristic* model that (i) assigns a line/block boundary to a rest based on its duration with the same probability, as reported in Figure 4, and (ii) fills the space between any two boundaries with lyrics of the appropriate syllable counts. This *Heuristic* model computes the following word probability:

$$P(w_t|w_0, ..., w_{t-1}, \mathbf{m}) = \tag{8}$$

$$\begin{cases} Q(\langle\text{BOB}\rangle|m_{i(t+1)}) & (\text{if } w_t = \langle\text{BOB}\rangle) \\ Q(\langle\text{BOL}\rangle|m_{i(t+1)}) & (\text{if } w_t = \langle\text{BOL}\rangle) \\ (1 - Q(\langle\text{BOB}\rangle|m_{i(t+1)}) - Q(\langle\text{BOL}\rangle|m_{i(t+1)}))\times \\ \frac{P_{\text{LSTM}}(w_t|w_0,...,w_{t-1})}{1-P_{\text{LSTM}}(\langle\text{BOL}\rangle|w_0,...,w_{t-1})-P_{\text{LSTM}}(\langle\text{BOB}\rangle|w_0,...,w_{t-1})} \\ & (\text{otherwise}) \end{cases}$$

where Q is the same probability as reported in Figure 4. P_{LSTM} is the word probability calculated by a standard LSTM language model. The remaining two are Melody-conditioned RNNLMs with the proposed learning strategies: (5) *Fine-tuned* and (6) *Pseudo-melody* models.

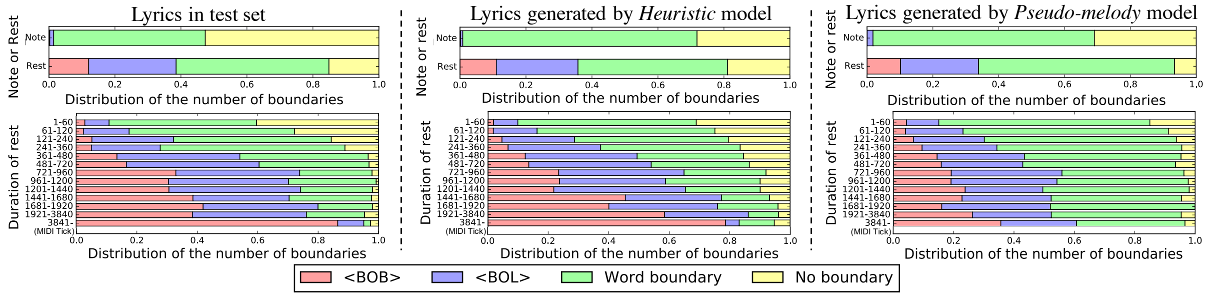

Figure 7: Distribution of the number of boundaries in the test set and lyrics generated by the *Heuristic* and *Pseudo-melody* models.

The top part of Table 1 summarizes the performance of these models. Regarding the boundary replication, the *Heuristic*, *Alignment-only*, *Fine-tuned*, and *Pseudo-melody* models achieved higher performance than the *Lyrics-only* model for unlabeled matching of line/block boundaries (i.e., UB). This result indicates that our Melody-conditioned RNNLMs successfully capture the consistency between melody and boundaries of lyrics. The results of the *Full-data* model is low (as expected) because the size of the melody-lyrics alignment data is far smaller than that of the raw lyrics data and this harms the learning process of the dependency between melody and lyrics. For the block boundary, the *Heuristic* model achieved the best performances. For the line boundary, the *Fine-tuned* model achieved the best performances.

Regarding PPL and PPL-W, the *Lyrics-only*, *Full-data*, and *Pseudo-melody* models show better results than the other models. The *Fine-tuned* model shows reduced performance compared with the *Lyrics-only* model because fine-tuning with a small amount of data causes overfitting in the language model. Also, the training size of the *Alignment-only* model is insufficient for learning a language model of lyrics. Interestingly, the *Pseudo-melody* model achieved better performance than the *Full-data* model and overall achieved the best score. This result indicates that the *Pseudo-melody* model uses the information of a given melody to make a better prediction of its lyrics word sequence. On the other hand, the *Heuristic* model had the worst performance, despite training with a large amount of raw lyrics. We analyze the reason for such performance and describe our results in Section 5.5. It is not necessarily clear which to choose, either the *Fine-tuned* or *Pseudo-melody* model, which may depend also on the size and diversity of the training and test data. However, one can conclude at least that combining a limited-scale collection of melody-lyrics alignment data with a far larger collection of lyrics-alone data boosts the model's capability of generating a fluent lyrics which structurally fits well the input melody.

5.4 Effect of predicting syllable-counts

To investigate the effect of predicting syllable-counts, we compared the performance of the proposed models to models that exclude the syllable-count output layer $\mathbf{y}_s$. The middle part of Table 1 summarizes the results. For the pretraining strategy, the use of $\mathbf{y}_s$ successfully alleviates data sparsity when learning the correlation between syllable counts and melodies from only words themselves. As can be seen, the model without $\mathbf{y}_s$ shows reduced performance relative to both PPLs and the boundary replication. On the other hand, for the pseudo-melody strategy, the two models are competitive in both measures. This means that the *Pseudo-melody* model obtained a sufficient amount of word-melody input pairs to learn the correlation.

5.5 Analysis of melody and generated lyrics

To examine whether the models can capture correlations between rests and boundaries of lyrics, we calculate the proportion of the word, line, and block boundaries in the original lyrics and in the lyrics generated by the *Heuristic* and *Pseudo-melody* model for the test set (Figure 7). The proportion of ⟨BOL⟩ and ⟨BOB⟩ generated by the *Heuristic* model are almost equivalent to those of the original lyrics. On the other hand, for the *Pseudo-melody* model, the proportion of line/block boundary types for the longer rests are smaller than that of the original lyrics.

Although the *Heuristic* model reproduces the proportion of the original line/block boundaries, the model had a low performance in terms of PPL,

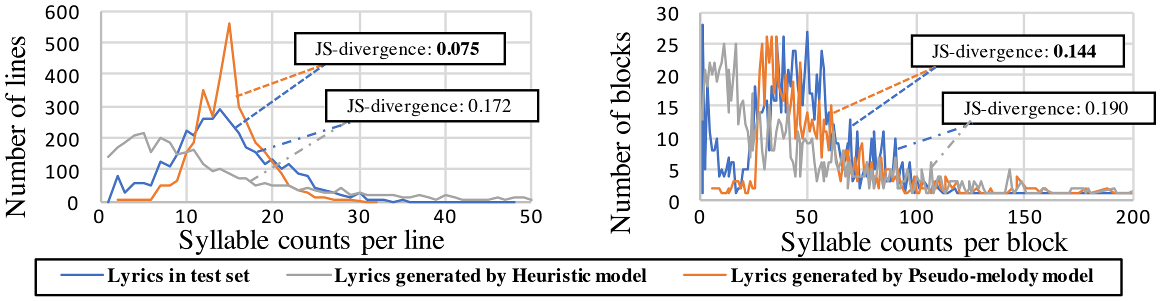

Figure 8: Distribution of the syllable count of the generated lines/blocks

	Heuristic		*Lyrics-only*		*Fine-tuned*		*Pseudo-melody*		*Human* (Upper-bound)	
Measure	Means ± SD	Median	Means ± SD	Median	Means ± SD	Median	Means ± SD	Median	Means ± SD	Median
L	2.06±1.08	2	2.33±1.23	2	**2.85**±1.20	3	**2.93**±1.14	3	3.56±1.33	4
G	2.28±1.07	2	**2.81**±1.16	3	2.79±1.06	3	**2.97**±1.08	3	3.50±1.25	4
LM	2.34±1.07	2	**2.91**±1.15	3	2.70±1.13	3	**2.96**±1.09	3	3.49±1.35	4
DM	2.33±1.10	2	**2.80**±1.06	3	2.59±1.11	3	**2.89**±1.07	3	3.49±1.30	4
OQ	2.01±1.01	2	2.59±1.15	3	2.42±1.08	2	**2.65**±1.01	3	3.32±1.19	4

Table 2: Results of the qualitative evaluation.

as shown in Section 5.3. By investigating the lyrics generated by the *Heuristic* model, we found that the model tends to generate line/block boundaries after the melody rest, even if the two rests are quite close. Figure 8 shows the distributions of the syllable per line / block frequency and the distributions of the Jensen-Shannon divergence. While the Heuristic model tends to generate short lines/blocks, our model generates the lyrics so that lines/blocks do not become too short. This result supports that (i) our model is trained using melody and lyric contexts and (ii) the heuristic approach, which simply generates line/block boundaries based on the distribution in Figure 4, cannot generate fluent lyrics with well-formed line/block lengths.

6 Qualitative evaluation

To asses the quality of the generated lyrics, inspired by (Oliveira, 2015), we asked 50 Yahoo crowdsourcing workers to answer the following five questions using a five-point Likert scale:

Listenability (L) When listening to melody and lyrics, are the positions of words, lines, and segments natural? (1=*Poor* to 5=*Perfect*)

Grammaticality (G) Are the lyrics grammatically correct? (1=*Poor* to 5=*Perfect*)

Line-level meaning (LM) Is each line in the lyrics meaningful? (1=*Unclear* to 5=*Clear*)

Document-level meaning (DM) Are the entire lyrics meaningful? (1=*Unclear* to 5=*Clear*)

Overall quality (OQ) What is the overall quality of the lyrics? (1=*Terrible* to 5=*Great*)

For the evaluation sets, we randomly selected four melodies from the RWC Music Database (Goto et al., 2002). For each melody, we prepared four lyrics generated by the *Heuristic*, *Lyrics-only*, *Fine-tuned*, and *Pseudo-melody* models. Moreover, to obtain an upper bound for this evaluation, we used the lyrics created by amateur writers: we asked four native Japanese speakers to write lyrics on the evaluation melody. One writer was a junior high school teacher of music who had experience in music composition and writing lyrics. Three writers were graduate students with different levels of musical expertise. Two of the three writers had experience with music composition, but none of them had experience with writing lyrics.[9] As a result, we obtained 50 (workers) × 4 (melodies) × 5 (lyrics) samples in total. We note that workers did not know whether lyrics were created by a human or generated by a computer.

Table 2 shows the average scores, standard deviations, and medians for each measure. Regarding the "Listenability" evaluation, workers gave high scores to the *Fine-tuned* and *Pseudo-melody* models that are trained using both the melody and lyrics. This result is consistent with the perplexity evaluation result. On the other hand, regarding the "Grammaticality" and "Meaning" evaluation, workers gave high scores to the *Lyrics-only* and *Pseudo-melody* models that are well-trained on a large amount of text data. This result is consistent with the result of

[9] We release lyrics and audio files used in the qualitative evaluation on the Web (https://github.com/KentoW/deep-lyrics-examples).

the boundary replication task. Regarding the "Overall quality" evaluation, the *Pseudo-melody* model outperformed all other models. These results indicate our pseudo data learning strategy contributes to generating high-quality lyrics. However, the quality of lyrics automatically generated is still worse than the quality of lyrics that humans produce, and it still remains an open challenge for future research to develop computational models that generate high-quality lyrics.

7 Related work

In the literature, a broad range of research efforts has been reported for computationally modeling lyrics-specific properties such as meter, rhythm, rhyme, stress, and accent Greene et al. (2010); Reddy and Knight (2011); Watanabe et al. (2014, 2016). While these studies provide insightful findings on the properties of lyrics, none of those takes the approach of using melody-lyrics parallel data for modeling correlations of lyrics and melody structures. One exception is the work of Nichols et al. (2009), who used melody-lyrics parallel data to investigate, for example, the correlation between syllable stress and pitch; however, their exploration covers only correlations at the prosody level but not structural correlations.

The same trend can be seen also in the literature of automatic lyrics generation, where most studies utilize only lyrics data. Barbieri et al. (2012) and Abe and Ito (2012) propose a model for generating lyrics under a range of constraints provided in terms of rhyme, rhythm, part-of-speech, etc. Potash et al. (2015) proposes an RNNLM that generates rhymed lyrics under the assumption that rhymes tend to coincide with the end of lines. In those studies, the melody is considered only indirectly; namely, input prosodic/linguistic constraints/preferences on lyrics are assumed to be manually provided by a human user because the proposed models are not capable of interpreting and transforming a given melody to constraints/preferences.

For generating lyrics for a given melody, we have so far found in the literature two studies which propose a method. Oliveira et al. (2007) and Oliveira (2015) manually analyze correlations among melodies, beats, and syllables using 42 Portuguese songs and propose a set of heuristic rules for lyrics generation. Ramakrishnan A et al. (2009) attempt to induce a statistical model for generating melodic Tamil lyrics from melody-lyrics parallel data using only ten songs. However, the former captures only phonological aspects of melody-lyrics correlations and can generate a small fragment of lyrics (not an entire lyrics) for a given piece of melody. The latter suffers from the severe shortage of data and fails to conduct empirical experiments.

8 Conclusion and future work

This paper has presented a novel data-driven approach for building a melody-conditioned lyrics language model. We created a 1,000-song melody-lyrics alignment dataset and conducted a quantitative investigation into the correlations between melodies and segment boundaries of lyrics. No prior work has ever conducted such a quantitative analysis of melody-lyrics correlations with this size of data. We have also proposed a RNN-based, melody-conditioned language model that generates fluent lyrics whose word/line/block boundaries fit a given input melody. Our experimental results have shown that: (1) our Melody-conditioned RNNLMs capture the consistency between melody and boundaries of lyrics while maintaining word fluency; (2) combining a limited-scale collection of melody-lyrics alignment data with a far larger collection of lyrics-alone data for training the model boosts the model's competence; (3) we have also produced positive empirical evidence for the effect of applying a multi-task learning schema where the model is trained for syllable count prediction as well as for word prediction; and (4) the human judgments collected via crowdsourcing showed that our model improves the quality of generated lyrics.

For future directions, we plan to further extend the proposed model for capturing other aspects of lyrics/melody discourse structure such as repetitions, verse-bridge-chorus structure, and topical coherence of discourse segment. The proposed method for creating melody-lyrics alignment data enables us to explore such a broad range of aspects of melody-lyrics correlations.

Acknowledgments

This study utilized the RWC Music Database (Popular Music). This work was partially supported by a Grant-in-Aid for JSPS Research Fellow Grant Number JP16J05945, JSPS KAKENHI Grant Numbers JP15H01702, and JST ACCEL Grant Number JPMJAC1602. The authors would like to thank Dr. Paul Reisert for the English language review.

References

Chihiro Abe and Akinori Ito. 2012. A Japanese lyrics writing support system for amateur songwriters. In *Proceedings of Asia-Pacific Signal & Information Processing Association Annual Summit and Conference 2012 (APSIPA ASC 2012)*. pages 1–4.

Dave Austin, Jim Peterik, and Cathy Lynn Austin. 2010. *Songwriting for Dummies*. Wileys.

Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. 2012. Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*. pages 115–120.

Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. 2002. RWC Music Database: Popular, classical and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*. volume 2, pages 287–288.

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*. pages 524–533.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*. volume 37, pages 448–456.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint* arXiv:1412.6980.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*. pages 230–237.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech 2010*. pages 1045–1048.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48(3):443–453.

Eric Nichols, Dan Morris, Sumit Basu, and Christopher Raphael. 2009. Relationships between lyrics and melody in popular music. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*. pages 471–476.

Hugo Gonçalo Oliveira. 2015. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence* 6(1):87–110.

Hugo R. Gonçalo Oliveira, F. Amialcar Cardoso, and Francisco C. Pereira. 2007. Tra-la-lyrics: An approach to generate text based on rhythm. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*. pages 47–55.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for automatic Rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. pages 1919–1924.

Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi. 2009. Automatic generation of Tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*. pages 40–46.

Sravana Reddy and Kevin Knight. 2011. Unsupervised discovery of rhyme schemes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2011)*. pages 77–82.

Tatsuji Ueda. 2010. よくわかる作詞の教科書 *The writing lyrics textbook which is easy to understand (in Japanese)*. YAMAHA music media corporation.

Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. 2014. Modeling structural topic transitions for automatic lyrics generation. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation (PACLIC 2014)*. pages 422–431.

Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. 2017. LyriSys: An Interactive support system for writing lyrics based on topic transition. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces (ACM IUI 2017)*. pages 559–563.

Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. 2016. Modeling discourse segments in lyrics using repeated patterns. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. pages 1959–1969.