Modeling Discourse Structure of Lyrics

Kento Watanabe

Graduate School of Information Sciences

Tohoku University

A thesis submitted for the degree of

Doctor of Information Science

January 2018

Acknowledgements

I would like to thank Professor Kentaro Inui whose comments and suggestions were of inestimable value for my whole study. He is the chief examiner for my thesis and has made his support available in a number of ways. I also would like to express my gratitude to Professor Akinori Itoh and Professor Satoshi Shioiri. They are the sub examiner of my thesis. They gave me a lot of valuable comments for improving the quality of my study.

I am also thank to Professor Naoaki Okazaki at Tokyo Institute of Technology. He gives me constructive comments and warm encouragement. I am particularly grateful for the assistance given by Assistant Professor Yuichiroh Matsubayashi. His support and encouragement were invaluable.

I am deeply grateful to Dr. Masataka Goto, Dr. Satoru Fukayama, and Dr. Tomoyasu Nakano at National Institute of Advanced Industrial Science and Technology (AIST) in Japan. When I had been a visiting researcher at AIST for six times from 2014 to 2017, they gave me a lot of useful and insightful comments. I believe that a lot of achievements in this thesis would not have been possible if I couldn ' t collaborate with them.

I gratefully appreciate the financial support of Japan Society of Promotion Sciences Fellowship that made it possible to complete my thesis.

My colleagues gives me constructive comments and warm encouragement. Especially, I would like to thank Yamaki-san, Narita-san, Sugawara-san, Inoue-san, and Aizawas-san the secretary of our lab. I believe that our lab would go bankrupt if they were not the secretary of our lab.

I would like to express the deepest gratitude to Hatsune Miku. I could always try hard because of her. Without her encouragement, this thesis would not have materialized.

Finally, I would like to express my gratitude to my family for their kind support and warm encouragements.

Abstract

Lyrics are an important element in popular music that conveys stories and expresses emotion. Unlike prose text, writing lyric requires consideration of music-specific properties such as the structure of melody notes and rests, rhythms, and repetitions. Especially, lyrics also have paragraphs and sentences corresponding to *discourse segments*. Thus, the goal of this thesis is modeling the discourse nature of lyrics to understand the structure of lyrics.

Once a reasonably sophisticated computational model of discourse structure of lyrics is obtained, the model will provide us a better understanding of the nature and structure of lyrics, which will then allow us to consider building computer systems which can enhance the creativity of human lyrics writers. In spite of its importance, however, no prior study has ever addressed the issue of modeling this discourse-oriented nature of lyrics.

This is the first study that takes a data-driven approach to exploring the discourse structure of lyrics in relation to various lyrics-specific properties. In particular, we propose several computational models of discourse structure from three viewpoints.

- The first viewpoint is modeling discourse segments in lyrics. This thesis conduct the first large-scale corpus study on discourse segments in lyrics, where we examine our primary hypothesis that discourse segmentations in lyrics strongly correlate with repeated patterns. To test our hypothesis that discourse segmentations in lyrics strongly correlate with repeated patterns, we conduct the first large-scale corpus study. Then, we propose the task to automatically identify segment boundaries in lyrics and train a logistic regression model for the task with the repeated pattern and textual features. The results

of our experiments illustrate the significance of capturing repeated patterns in predicting the boundaries of discourse segments in lyrics.

- The second viewpoint is modeling the two common discourse-related notions: storylines and themes. We assume that a storyline is a chain of transitions over topics of segments and a song has at least one entire theme. We then hypothesize that transitions over topics can be captured by a probabilistic topic model which incorporates a distribution over transitions of latent topics and that such a distribution of topic transitions is affected by the theme of lyrics. To test those hypotheses, we conduct experiments on the word and segment order prediction tasks. The experimental results show that our probabilistic topic model capture storylines effectively and considering the notion of theme contributes to the modeling of storylines.
- The third viewpoint is modeling relationship between melodies and discourse segments. To investigate this relationship, we created a collection of 1,000 lyrics-melody pairs augmented with mora-note alignments and word/sentence/paragraph boundaries, which is, to our knowledge, the biggest collection of precisely aligned lyrics-melody parallel data provided in this field. We report on our quantitative analysis of the correlation between word/sentence/segment boundaries in lyrics and rests in melodies. Our experimental results show strong tendency that the boundaries of larger segments tend to coincide more with longer rests.

In addition to the above studies, this thesis presents a novel, purely data-driven model for lyrics generation by conditioning an language model with a featurized input constraints such as topics, mora-counts, and melodies. Our experimental results show that combining a limited-scale collection of lyrics-melody alignment data and a far larger collection of lyrics-alone data for training the model boosts the model's capability of generating a fluent lyrics and fitting it to the input melody.

We also propose a novel lyric-writing support system using our lyrics language model. Previous systems for lyric writing can fully automatically

only generate a single line of lyrics that satisfies given constraints on accent and syllable (or mora) patterns or an entire lyric. In contrast to such systems, our system allows users to create and revise their study incrementally in a trial-and-error manner. Through fine-grained interactions with the system, the user can create the specifications of the discourse structure, mora-counts, and most importantly, storylines (i.e., the transition over topics of segments).

This thesis is a basic research of modeling discourse structure of lyrics and a practical research for computer-assisted or fully-automated creation of lyrics.

Contents

List of Figures

Chapter 1

Introduction

Lyrics are an important element of popular music. They provide an effective means to express the message and emotion of music. Unlike prose text, lyrics have their own peculiar properties, such as the structure of melody notes and rests, rhythms, repetitions, etc. [Austin et al., 2010; Ueda, 2010]. A simple example is the correlation between word boundaries in lyrics and rests in melody. As in Figure 1.1, one feels it awkward if a single word spans beyond a (long) melody rest. A lyrics writer needs to consider such constraints in content and lexical selection, which imposes extra cognitive loads. The writer may also consider using rhymes, refrains or repetitions to color the entire story rhetorically as in the example lyrics shown in Figure 1.2, where rhymes can be seen at *night*, *light* and *tight* in Segment 7. Writing lyrics is thus a complex task.

These properties of lyrics have been motivating a range of research for computer-based modeling of lyrics [Greene et al., 2010; Mayer et al., 2008; Nichols et al., 2009; Reddy and Knight, 2011] and computer-assisted or fully-automated creation of lyrics [Abe and Ito, 2012; Barbieri et al., 2012; Ghazvininejad et al., 2016; Oliveira et al., 2007; Potash et al., 2015; Ramakrishnan A et al., 2009]. In particular, building a computational model of lyrics is an important research goal. Once a reasonably sophisticated computational model of lyrics is obtained, the model will provide us a better understanding of the nature and structure of lyrics, which will then allow us to consider building computer systems which can enhance the creativity of human lyrics writers. In reality, however, while an increasing number of papers have been published for demonstrating computer systems that automatically generate lyrics or assist human lyricists [A and Devi, 2010; Abe and Ito, 2012; Barbieri et al., 2012; Ghazvininejad

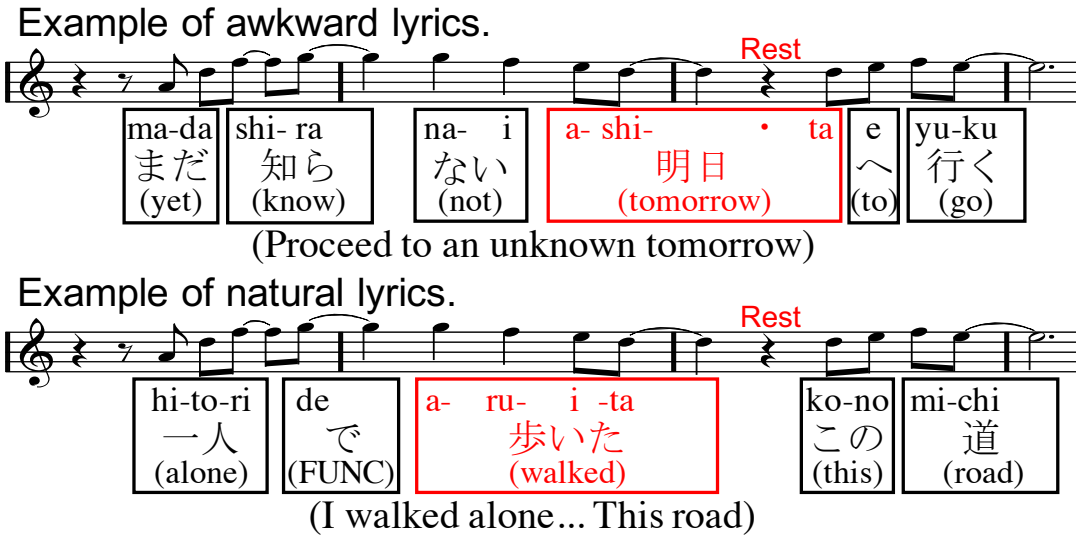


Figure 1.1: Examples of awkward and natural lyrics. The translated English lyrics are given in parentheses. (FUNC) indicates a function word. The song is from the RWC Music Database (RWC-MDB-P-2001 No.20) [Goto et al., 2002].

et al., 2016; Oliveira et al., 2007; Potash et al., 2015; Wu et al., 2013], research for modeling lyrics and understanding their properties is still limited [Greene et al., 2010; Mayer et al., 2008; Nichols et al., 2009; Reddy and Knight, 2011].

One crucial issue we miss in those studies is modeling the nature of lyrics as *discourse*. Similar to prose text, a piece of lyrics typically comprises paragraphs and sentences corresponding to discourse segments; namely, lyrics of popular music typically has the *verse*, *bridge*, and *chorus* segments [Mahedero et al., 2005] and such segments may comprise more fine-grained segments as in Figure 1.2. Each segment provides part of the entire story and the segments are organized (or sequentially ordered) so as to constitute a coherent structure as a whole. Moreover, chorus segments appear repeatedly (e.g., in Figure 1.2, segments 5, 8 and 9 are identical to segment 1), which is not typically observed in prose text. In spite of its importance, however, no prior study has ever addressed the issue of modeling this discourse-oriented nature of lyrics.

Motivated by this background, in this thesis, we report on our novel study for building several computational models of the discourse nature of lyrics.

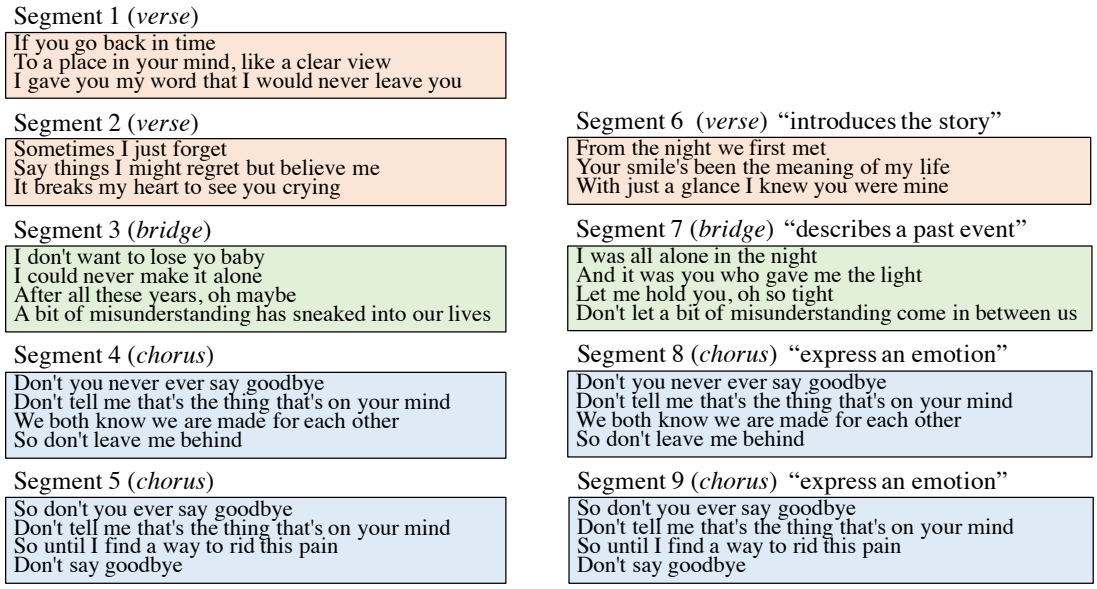


Figure 1.2: An example of lyrics with a storyline and repeated patterns (title: *Don't Say Good bye* (RWC-MDB-P-2001 No.90 from RWC Music Database [Goto et al., 2002])).

1.1 Research Issues

In this thesis, we address following four research issues.

Does the discourse segments in lyrics strongly correlate with repeating patterns? Phrases of lyrics often appear repeatedly, and this repeated pattern may be correlated with discourse segments. For example, if a sequence of lines in lyrics has a repetition, such as *abcdefabc* (each letter represents a line, with repeated letters being repeated lines), we expect the boundaries of the discourse segments tend to agree with the boundaries of the repeated parts as in $|abc|def|abc|$, where "|" indicates a boundary. However, no prior study has ever verified this correlation.

What is the most suitable way to model storylines in lyrics? Each discourse segment in lyrics provides part of the entire story and the segments are organized (or sequentially ordered) so as to constitute a coherent structure as a whole. In Figure 1.2, for example, Segment 6 introduces the story, Segment 7 retrospects a past event, and Segment 8 and 9 express an emotion which arises from the retrospection. However, no study has ever addressed the issue of modeling storylines

in lyrics.

Does the discourse segments in lyrics strongly correlate with melody? Several correlations between melody and lyrics are expected. For example, words, sentences and segments rarely span beyond a long melody rest and the boundaries of larger components (i.e., discourse segments) tend to coincide more with longer rests. This direction of research, however, has never been promoted partly because it requires a large training dataset consisting of aligned pairs of lyrics and melody but so far no such data has been available for research. We refer to the data as a *melody-lyric alignment* data.

Are discourse structure models efficient in automatic lyrics generation task? In addition to modeling the discourse structure, we are interested in the effectiveness of discourse model for demonstrating computer systems that automatically generate lyrics or assist human lyricists.

1.2 Contributions

This thesis makes following contributions.

Modeling and investigating discourse segments in lyrics. To examine our primary hypothesis that *discourse segments in lyrics strongly correlate with repeated patterns*, we consider the task of computationally predicting the boundaries of discourse segments in lyrics under the assumption that a better prediction model would allow us to better understand the nature of the discourse structure of lyrics. This is the first study that takes a data-driven approach to exploring the discourse structure of lyrics in relation to repeated patterns.

Modeling and investigating storylines in lyrics. To capture the two common discourse-related notions: storylines and themes, we hypothesize that transitions over topics of lyric segments can be captured by a probabilistic topic model which incorporates a distribution over transitions of latent topics and that such a distribution of topic transitions is affected by the theme of lyrics.

Melody-lyric alignment data. To create a large collection of lyrics-melody alignment data, we propose a methodology for creating melody-lyrics alignment data

by leveraging lyrics and their corresponding musical score data on the web. To our knowledge, this dataset is the biggest collection of precisely aligned lyrics-melody parallel data provided in this field. While this thesis reports on our experiments with Japanese song data, presumably the approach will work for other languages as well.

Modeling and investigating relationships between lyrics and melodies. We deeply analyze the correlation between melody and lyrics, and evaluate proposed model quantitatively. This is the first study that has ever provided such strong empirical evidence to the hypotheses about the correlations between lyrics segments and melody rests. In addition to this analysis, we propose a novel, purely lyrics generation model that output lyrics for an entire input melody. We extend a common Recurrent Neural Network Language Model (RNNLM) [Mikolov et al., 2010] so that its output words can be conditioned with a featurized input melody. To our knowledge, these are the first language models that jointly learn the consistency between word, line, and segment boundaries and melodies.

Novel interactive support system for writing lyrics. As mentioned in the beginning of this chapter, for writers of lyrics, considering various lyrics-specific properties simultaneously is not easy. This is where appropriate human-computer interaction system can reduce human loads. Therefore, this thesis provides an overview of the design of the system and its user interface and describes how the writing process is guided by our probabilistic discourse structure model. Our system can assist a writer in incrementally taking the above factors into account through an interactive interface by generating candidate pieces of lyrics that satisfy the specifications provided by the writer. The capability of automatically generating lyrics and allowing the user to create lyrics incrementally in a trial-and-error manner can be useful for both novices and experts.

1.3 Thesis Overview

The rest of this thesis is structured as follows.

Chapter 2: Natural Language Processing of Lyrics. In this chapter, we give an overview

of natural language processing of lyrics. We show that most of the previous studies focus on modeling lyric-specific properties except discourse structure.

Chapter 3: Modeling Discourse Segments Using Repeated Patterns. This chapter presents the first quantitative analysis of the distribution of repeated lines and segments in lyrics and suggests cues that could help to identify the segment boundaries. Then, we describes our computational model, which predicts the boundaries of discourse segments in lyrics using repeated patterns.

Chapter 4: Modeling Storylines. To investigate the effects of capturing topic transitions, we describe a probabilistic topic model which incorporates a distribution over transitions of latent topics. Moreover, to verify that both theme and topic transition are useful for modeling the storyline, we propose two extended combination models to handle theme and storyline simultaneously.

Chapter 5: Modeling Relationship between Melody and Lyrics. We first propose a novel method for creating a lyrics-melody aligned parallel dataset. We then quantitatively analyze the correlations between lyrics segment boundaries and melody rests. Finally, we build a lyrics language model capable of generating fluent lyrics whose segment boundaries fit a given input melody.

Chapter 6: Interactive Support System for Writing Lyrics. We describe the functions of the proposed system and then describe the probabilistic generative model which the system employs to generate candidate lyrics. Moreover, to investigate the capabilities, limitations, and potential of our interaction design, we asked human-users to use our system and collected preliminary user feedback.

Chapter 7: Conclusions. We summarize our discussion, and present our future direction.

Chapter 2

Natural Language Processing of Lyrics

This chapter briefly summarizes the related work on the natural language processing for lyrics. We first, describe the overview of modeling lyrics-specific properties such as rhyme, melody, and so on. We then, introduce overview of application tasks using lyrics processing.

2.1 Overview of Modeling Lyrics-specific Property

These prior studies share the motivation of modeling lyric-specific properties with our study. However, no previous study has ever considered capturing the discourse-oriented nature of lyrics whereas our study aims at modeling (1) discourse segments, (2) storylines, and (3) relationship between melody and discourse.

2.1.1 Modeling Styles of Lyrics

Several studies for capturing lyric-specific styles have been reported, where a broad range of music elements including meter, rhythm, rhyme, stressed/unstressed syllables, and accent are studied. Mayer et al. [2008] trained a support vector machine to classify music genres using only textual features such as rhyme and part-of-speech patterns. Greene et al. [2010] employed a finite-state transducer to assign syllable-stress pattern to all words in each line. Reddy and Knight [2011] developed a language-independent rhyme model based on a Markov process that finds rhyme schemes.

2.1.2 Modeling Relationship between Melodies and Lyrics

Oliveira et al. [2007] analyze correlations among melodies, beats, and syllables using 42 Portuguese songs. Nichols et al. [2009] identified several patterns in the relationship between the lyrics and melody in popular music by measuring the correlation between textual salience and musical salience.

2.1.3 Modeling Semantic Structure of Lyrics

Several studies aim at modeling semantic structure of lyrics. Kleedorfer et al. [2008] classified lyrics according to topical clusters calculated using nonnegative matrix factorization [Xu et al., 2003]. Sasaki et al. [2014] visualized lyric clusters calculated using Latent Dirichlet Allocation (LDA) [Blei et al., 2003].

2.1.4 Modeling Discourse Structure of Lyrics

Previous computational work into lyrics discourse segmentation has focused on identifying the segment labels of lyrics that are already segmented. For example, the structure of lyrics can be represented using labels A–B–C–A–B in which each letter refers to a group of lines; e.g., A might represent a chorus that appears twice. Barate et al. [2013] proposed a rule-based method to estimating such structure labels of *segmented* lyrics.

2.2 Overview of Application Tasks Using Lyrics Processing

2.2.1 Automatic Lyrics Generation

The same trend can be seen also in the literature of automatic lyric generation, where most studies play only with lyrics data alone. Barbieri et al. [2012] proposed a model for generating a lyrics under a range of constraints provided in terms of rhyme, rhythm, part-of-speech, etc. Potash et al. [2015] proposed an Recurrent Neural Network language model that generates rhymed lyrics under the assumption that rhymes tend to coincide with the end of lines. In those studies, melody is considered only indirectly;

namely, input prosodic/linguistic constrains/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 constrains/preferences.

For generating lyrics from a given melody, we have so far found in the literature only two studies which propose a method . Oliveira et al. [2007] proposed a set of heuristic rules for lyrics generation according to their quantitative analysis of Portuguese music data. Ramakrishnan A et al. [2009] attempted to induce a statistical model for generating melodic Tamil lyrics from lyrics-melody parallel data using only ten songs. However, the former captures only extremely limited aspects of lyrics-melody 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.

2.2.2 Writing Support Interface

Existing support systems can be classified into two types, systems that can generate entire lyrics automatically [Abe and Ito, 2012; Oliveira, 2015; Settles, 2010] and tools designed to assist the user in searching for words that satisfy a query and usage examples from stored lyrics. [1]

We will describe details of these related research in each chapter and clarify the contribution of our approach.

[1] MasterWriter. `http://masterwriter.com/`

Chapter 3

Modeling Discourse Segments Using Repeated Patterns

Our goal is to reveal the discourse structure of lyrics in popular music by quantitatively analyzing a large-scale lyrics corpus. As a first but crucial step toward achieving this goal, this chapter explores the nature of the discourse structure of lyrics with a focus on the repeated patterns as an indicator of segment boundaries. In particular, we examine our primary hypothesis that *discourse segments in lyrics strongly correlate with repeated patterns* as mentioned in chapter 1. Moreover, we consider the task of computationally predicting the boundaries of discourse segments in lyrics under the assumption that a better prediction model would allow us to better understand the nature of the discourse structure of lyrics.

3.1 Overview of Text Segmentation

This section reviews related work into the discourse structure of lyrics, with particular focus on the segmentation of text using repeated patterns.

Text segmentation is a classic text retrieval problem, and there exists a rich body of research into text segmentation in natural language processing. Various linguistic cues have been suggested to identify text boundaries such as expressions that frequently appear at the end of segments [Beeferman et al., 1999], contextual/topical changes [Choi, 2000; Malioutov and Barzilay, 2006; Riedl and Biemann, 2012], and word/entity rep-

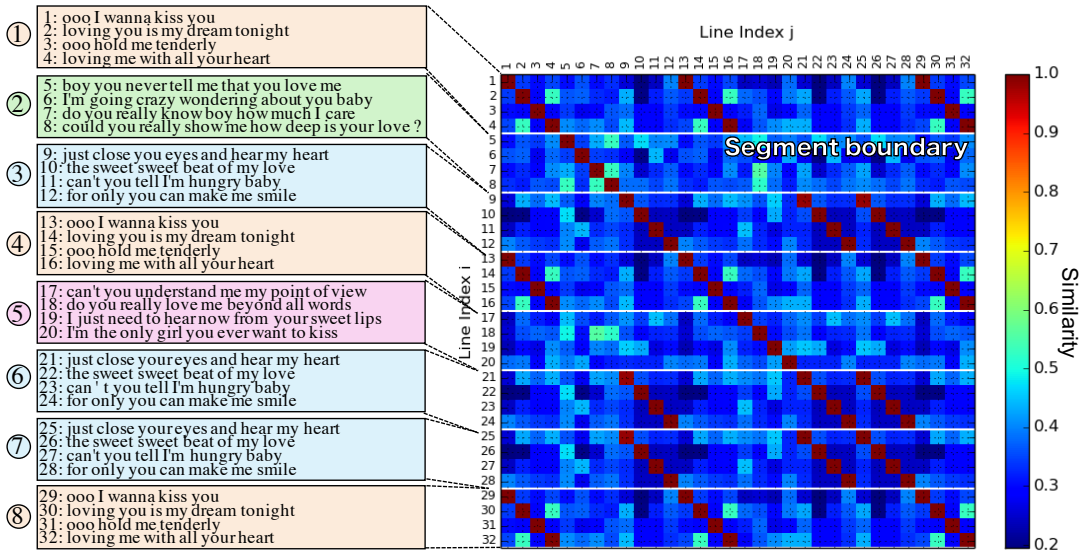


Figure 3.1: An example of lyrics and corresponding self-similarity matrix (title of lyrics: *How Deep Is Your Love?* (RWC-MDB-P-2001 No. 81 from RWC Music Database Goto et al. [2002]))

etition [Kan et al., 1998; Reynar, 1999].

Although we share the same motivation as these studies, these text segmentation methods do not consider repeated patterns of phrasal segments because this type of repetition is nearly always absent in prose text. On the other hand, segments in lyrics often have repetitions [Austin et al., 2010; Ueda, 2010] as shown in Section 3.2.1. We aim to capture the segment structure of lyrics using repeated patterns.

Previous computational work into lyrics segmentation has focused on identifying the segment labels of lyrics that are already segmented. For example, the structure of lyrics can be represented using labels A–B–C–A–B in which each letter refers to a group of lines; e.g., A might represent a chorus that appears twice. Barate et al. [2013] proposed a rule-based method to estimating such structure labels of *segmented* lyrics. Our task differs from this task in that we aim to estimate the segment boundaries of *unsegmented* lyrics using machine learning techniques.

In contrast to the segmentation of lyrics, much previous study has analyzed and estimated the segment structure of music audio signals using repeated musical parts such as verse, bridge, and chorus [Foote, 2000; Goto, 2006; Lu et al., 2004; McFee and Ellis, 2014; Paulus and Klapuri, 2006]. To automatically identify these repeated musical

parts in music audio signals, a self-similarity matrix (SSM) as shown in Figure 3.1 is often used. Repeated segments lead to high-valued lines in the off-diagonals of the matrix, and these patterns are used to identify the structure. To capture segments in lyrics using repeated patterns, we apply the SSM to lyrics. Lyrical repetition is known to be an important property of lyrics [Austin et al., 2010; Ueda, 2010], and we expect that repetition patterns would also appear in lyrics as they do in audio signals.

In summary, no previous computational work has exactly focused on the segmentation of lyrics using repeated patterns.

3.2 Statistics of Repeated Patterns and Segment Boundaries

As an initial step toward modeling the discourse structure of lyrics, we examine the distribution of segments in lyrics by focusing on repeated patterns. We first show the basic distributions of lyrics and suggest potential cues to indicate segment boundaries in lyrics. To examine the distribution of repeated patterns in lyrics and their relation to segment boundaries, we use a large scale lyrics database that contains 144,891 songs[1].

One issue to be addressed before conducting the corpus study is that no existing corpus has annotated the study is that none of existing corpus has annotation of discourse structure of lyrics. In this study, we preliminarily assume that discourse segment boundaries are indicated by empty lines inserted by lyrics writers. We admit that empty lines may not be "true" discourse segment boundaries and discourse segments may exhibit a hierarchical structure (e.g., verse–bridge–chorus structure). These issues could be better addressed by combining the analysis of the discourse structure of lyrics with the structural analysis of music. We believe this direction of research will open an intriguing new field for future exploration.

3.2.1 The Basic Distribution of Lyrics

Among the 144,891 songs in the lyrics database, there are 5,666,696 lines and 969,176 segments in total, with segment breaks inferred from empty lines. Per song, there are

[1]Music Lyrics Database. `http://www.odditysoftware.com/page-datasales1.htm`

39.11 lines and 6.69 segments on average. Most songs have at least one repeated line (84.79% using an exact criterion; 90.34% using a lenient matching criterion of normalized edit distance ≥ 0.8, explained in the next section). A fair number of songs also have at least one repeated segment (exact match: 37.73%, lenient match: 54.57%). Per song, 13.73 lines and 0.52 segments (both exact match) are repeated at least once on average. These distributions show that repetition of lines and segments occurs frequently in lyrics, in line with our expectations. Next, we suggest potential repeated patterns to help in identifying segment boundaries.

3.2.2 Correlation between Repeated Patterns and Segment Boundaries

To examine what kinds of repeated patterns would help identify segments in lyrics, we use the SSM, similar to previous study into the segmentation of music audio signals (Section 4.1). Figure 3.1 shows an example SSM. Throughout this study, we represent the i^{th} line in lyrics as l_i ($1 \leq i \leq L$), where L is the number of lines of the lyrics. A degree of similarity between l_i and l_j, i.e., $\text{sim}(l_i, l_j)$, is represented as an intensity at a cell where the i^{th} row and j^{th} column overlap. Using the normalized edit distance $\text{NED}(l_i, l_j)$, we compute the degree of similarity [Yujian and Bo, 2007]: $\text{sim}(l_i, l_j) = 1 - \text{NED}(l_i, l_j)$. The red diagonal lines in Figure 3.1 are the result of exact line repetitions[1]. The white horizontal lines in Figure 3.1 indicate the true segment boundaries.

After manually examining more than 1,000 lyrics and their SSMs, we suggest the following four types of repeated patterns as indicators of segment boundaries.

(1) The Start and end points of a diagonal line are segment boundaries. Some repeated segments correspond to the red diagonal lines. For example, in Figure 3.1, segment ① is repeated twice (segments ④ and ⑧), and each repetition, starting at l_{13} and l_{29}, can be observed as a diagonal line from l_1 to l_4. This suggests that some segments could be divided at the start and end points of such a diagonal line.

(2) A segment boundary does not appear within a diagonal line. This is related to

[1]The diagonal line of $\text{sim}(l_i, l_i)$ is ignored for analysis because it conveys no information.

Table 3.1: Correlation between each repeated pattern and segment boundary

Repeated pattern	Prior and conditional probabilities	Value
Pattern (1)	P(Boundary appears)	0.1455 (824286/5666696)
	P(Boundary appears | at the starting/ending of a diagonal line)	**0.2218** (339020/1530824)
Pattern (2)	P(Boundary does not appear)	0.8545 (4842410/5666696)
	P(Boundary does not appear | within a diagonal line)	**0.9273** (751195/810098)
Pattern (3)	P(Adjacent lines appear within a segment)	0.8507 (4697520/5521806)
	P(Adjacent lines appear within a segment | adjacent lines are similar)	**0.9439** (218524/231518)
Pattern (4)	P(Boundary appears after l_i)	0.1455 (824286/5666696)
	P(Boundary appears after l_i | l_i is similar to the last line of lyrics)	**0.4230** (125659/297069)
	P(Boundary appears after l_i | l_{i+1} is similar to the first line of lyrics)	**0.4189** (46531/111079)

(1). A segment boundary does not normally appear within a diagonal line because each diagonal line often corresponds to a segment.

(3) Similar adjacent lines appear within a segment. Line-level repetitions that are adjacent, such as rhymes and refrains, tend to occur within a segment. For example, line l_7 rhymes internally with l_8 where these lines appear within a segment because $\mathrm{sim}(l_7, l_8)$ indicates moderate similarity.

(4) A line similar to the first or last line of a song is an indicator of a segment boundary. Lines similar to the first or last line of lyrics tend to be repeated at segment boundaries. For example, in Figure 3.1, the first and last lines of the song, i.e., l_1 and l_{32}, are exactly the same as the first and last lines of segments ④ and ⑧. This is because the first and last lines of lyrics tend to be part of a chorus section that is often repeated throughout the lyrics.

To examine the extent to which these four patterns correlate with segment boundaries, we compute the prior and conditional probabilities of each pattern using the full lyrics database. Table 3.1 shows that all conditional probabilities are greater than their corresponding prior probabilities. These results suggest that the above repeated patterns reasonably capture segment boundaries, supporting the use of repeated patterns for modeling the segment structure of lyrics.

Note that pattern (1) does not hold for many cases. Figure 3.2 illustrates a typical negative exam-

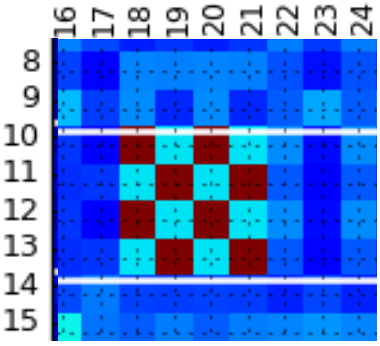


Figure 3.2: Negative example against pattern (1).

ple against pattern (1). This figure includes three diagonal lines, but the shorter lines do not agree with a segment boundary at either end. Similar cases are abundant partly because even a repetition of a single line is identified as a diagonal line in this experiment. This problem implies that a single occurrence of our local repeated pattern is not a sufficient clue for identifying a segment boundary. The conflict between patterns (1) and (2) is also shown in Figure 3.1, where a segment boundary implied by pattern (1) bisects a diagonal line from (l_{25}, l_9) to (l_{32}, l_{16}), which goes against pattern (2). This motivates us to build a machine learning-based model to capture combinations of multiple clues. The subsequent section describes how we represent these repeated patterns as features for predicting segment boundaries.

3.3 Computational Modeling of Segment Patterns in Lyrics

To confirm the validity of our four repeated patterns for segment structures, we address the novel task of detecting segment boundaries in lyrics. Given the lyrics of a song where all segments are concatenated (no empty lines), the task is to identify the segment boundaries of the lyrics reproducing the empty lines. We formalize this task as a binary classification problem to predict the end ($y = 1$) or continuation ($y = 0$) of a segment between lines l_i and l_{i+1}. We model the conditional probability $p(y|i)$ using logistic regression with two different types of features: (1) *repeated patterns in lyrics* and (2) *textual expressions appearing at the line boundaries*.

3.3.1 Repeated Patterns

We propose four subtypes of repeated pattern features (RPF1, RPF2, RPF3, and RPF4) corresponding to the four hypotheses presented in Section 3.2.2. Here, matrix M denotes the SSM of the lyrics. Each element $m_{i,j}$ represents the similarity between lines l_i and l_j, i.e., $m_{i,j} = \mathrm{sim}(l_i, l_j)$.

RPF1 The first repeated pattern (*the beginning or end point of a diagonal line in an SSM is a clue for a segment boundary*) is formalized as follows. Given two lines i and j, we expect that there exists a boundary after both of these lines if the lines are similar/dissimilar, but $i + 1$ and $j + 1$ are opposite (dissimilar/similar). For a given line

i, Equation 3.1 enumerates a set of lines j ($1 \leq j \leq L$) where there may be boundaries after line i and every line j:

$$g_\lambda(i) = \{j \mid (m_{i,j} - \lambda)(m_{i+1,j+1} - \lambda) < 0\} \tag{3.1}$$

Here, λ is a threshold for detecting similarity and dissimilarity. The left side of Figure 3.3 illustrates four likely boundaries for line $i = 24$ with the threshold $\lambda = 0.6$: $g_{0.6}(24) = \{8, 12, 20, 28\}$.

Using the function $g_\lambda(i)$, we define feature functions $f_\lambda^{(\mathrm{RPF1\#})}(i)$ and $f_\lambda^{(\mathrm{RPF1v})}(i)$ that assess how likely it is that line i is located at the beginning or end points of diagonal lines in the SSM:

$$f_\lambda^{(\mathrm{RPF1\#})}(i) = |g_\lambda(i)| \tag{3.2}$$

$$f_\lambda^{(\mathrm{RPF1v})}(i) = \frac{1}{|g_\lambda(i)|} \sum_{j \in g_\lambda(i)} |m_{i,j} - m_{i+1,j+1}| \tag{3.3}$$

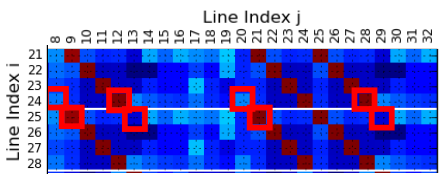
To sum up, $f_\lambda^{(\mathrm{RPF1\#})}(i)$ counts the number of likely boundaries after line i and other lines j, and $f_\lambda^{(\mathrm{RPF1v})}(i)$ computes the mean of the similarity differences at likely boundaries after line i and other lines j. We define multiple features with different threshold values λ.

RPF2 The second repeated pattern (*a segment boundary does not appear inside of a diagonal line of an SSM*) is formalized analogously to RPF1. Given two lines i and j, we expect that lines i and j are *points of continuity* if lines i and j are similar and $i+1$ and $j+1$ are also similar. For a given line i, Equation 3.4 enumerates a set of lines j ($1 \leq j \leq L$) where i and j are points of continuity:

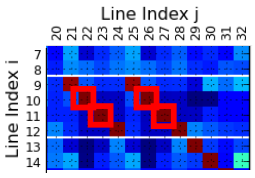
$$c_\lambda(i) = \{j \mid m_{i,j} \geq \lambda \wedge m_{i+1,j+1} \geq \lambda\} \tag{3.4}$$

The middle of Figure 3.3 shows an example of continuous points (here, $c_{0.6}(10) = \{22, 26\}$ in this example).

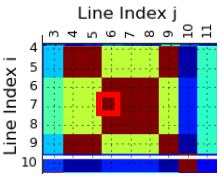
Similar to RPF1, Equations 3.5 and 3.6 count the number of continuous points and



Some diagonal lines end at line 24 and begin at line 25. A segment boundary is likely to appear between lines 24 and 25.



Two diagonal lines span over lines 9 to 12. Lines 9 to 12 are not likely to divide.



Lines 6 and 7 are similar. A segment boundary is unlikely to appear between lines 6 and 7.

Figure 3.3: Repeated pattern features: RPF1, RPF2, and RPF3.

Position	-3	-2	-1	0 (Line Break)	1	2	3
Word	oh	oh	!!		I	love	you
POS tag	UH	UH	SYM		PRP	VBP	PRP

6 words between i^{th} line and $i+1^{th}$ line

TF1_Uni-gram(-3) = "oh"	TF1_Bi-gram(-2) = "oh_oh"	TF1_Tri-gram(-2) = "oh_oh_!!"
TF1_Uni-gram(-2) = "oh"	TF1_Bi-gram(-1) = "oh_!!"	TF1_Tri-gram(-1) = "oh_!!_I"
TF1_Uni-gram(-1) = "!!"	TF1_Bi-gram(0) = "!!_I"	TF1_Tri-gram(1) = "!!_I_love"
TF1_Uni-gram(1) = "I"	TF1_Bi-gram(1) = "I_love"	TF1_Tri-gram(2) = "I_love_you"
TF1_Uni-gram(2) = "love"	TF1_Bi-gram(2) = "love_you"	
TF1_Uni-gram(3) = "you"		

Textual Feature 1 (TF1): 15 word N-grams

TF2_Uni-gram(-3) = "UH"	TF2_Bi-gram(-2) = "UH_UH"	TF2_Tri-gram(-2) = "UH_UH_SYM"
TF2_Uni-gram(-2) = "UH"	TF2_Bi-gram(-1) = "UH_SYM"	TF2_Tri-gram(-1) = "UH_SYM_PRP"
TF2_Uni-gram(-1) = "SYM"	TF2_Bi-gram(0) = "SYM_PRP"	TF2_Tri-gram(1) = "SYM_PRP_VBP"
TF2_Uni-gram(1) = "PRP"	TF2_Bi-gram(1) = "PRP_VBP"	TF2_Tri-gram(2) = "PRP_VBP_PRP"
TF2_Uni-gram(2) = "VBP"	TF2_Bi-gram(2) = "VBP_PRP"	
TF2_Uni-gram(3) = "PRP"		

Textual Feature 2 (TF2): 15 Part of speech N-grams

Figure 3.4: Textual features: TF1 and TF2.

the mean of the similarity differences at continuous points, respectively.

$$f_{\lambda}^{(\mathrm{RPF2\#})}(i) = |c_{\lambda}(i)| \tag{3.5}$$

$$f_{\lambda}^{(\mathrm{RPF2v})}(i) = \frac{1}{|c_{\lambda}(i)|} \sum_{j \in c_{\lambda}(i)} |m_{i,j} - m_{i+1,j+1}| \tag{3.6}$$

RPF3 (similarity with a subsequent line) RPF3 encodes the third repeated pattern, i.e., similar adjacent lines belong to the same segment. For a given line index i, this is quantified by the similarity $\mathrm{sim}(l_i, l_{i+1})$:

$$f^{(\mathrm{RPF3})}(i) = m_{i,i+1} \tag{3.7}$$

The right of Figure 3.3 shows an example where RPF3 indicates a continuation between lines 6 and 7.

RPF4 (similarity with the first and last lines) The fourth repeated pattern (i.e., *a line similar to the first line of the lyrics is likely to be the first line of a segment*, and *a line similar to the last line of the lyrics is likely to be the last line of a segment*) is encoded by two feature functions $f^{(\mathrm{RPF4b})}(i)$ and $f^{(\mathrm{RPF4e})}(i)$:

$$f^{(\mathrm{RPF4b})}(i) = m_{i,1} \tag{3.8}$$

$$f^{(\mathrm{RPF4e})}(i) = m_{i,n} \tag{3.9}$$

3.3.2 Textual Expressions

Some textual expressions appear selectively at the beginning or end of a segment. For example, the phrase "So I" often appears at the beginning of a line but rarely appears at the beginning of a segment. To exploit such indications of the beginnings/ends of lines, we propose two textual features (TF1 and TF2).

TF1 (word n-grams at a line boundary) A phrase like "oh oh !!" tends to appear at the end of a segment. In contrast, a phrase like "I'm sorry" may appear at the beginning of a segment. Previous study on sentence boundary estimation has often used n-grams to detect segment boundaries [Beeferman et al., 1999]. Thus, we define word n-gram features (for $n = 1, 2, 3$) around a line boundary. More specifically, we define 15 n-gram features at different positions, listed and illustrated with an example in Figure 3.4.

TF2 (part of speech n-grams around a line boundary) Parts of speech (POS), such as *particles* or *determiners* do not tend to appear at the end of a sentence, and *conjunctions* do not appear at the beginning of a sentence. We exploit these tendencies by defining features for POS. Similar to TF1, we define POS n-gram features (for $n = 1, 2, 3$) around a line boundary. Specifically, we define 15 POS n-gram features at different positions, as shown in Figure 3.4.

3.4 Experiment

We sampled 105,833 English songs from the Music Lyrics Database v.1.2.7 so that each song contains at least five segments. The resulting dataset includes 2,788,079 candidate boundaries and 517,234 actual boundaries. We then split these songs into training (60%), development (20%), and test (20%) sets. For feature extraction, we used the Stanford POS Tagger [Toutanova et al., 2003]. To train the segment boundary classifiers, we used the Classias implementation [Okazaki, 2009] of L2-regularized logistic regression. By employing multiple threshold values of λ from 0.1 to 0.9 with a step size of 0.1, we used them all together.

3.4.1 Performance Evaluation Metrics

We used two sets of metrics to evaluate the performance of each model for the task. One was standardly used in audio music segmentation, i.e., the precision, recall, and F-measure of identifying segment boundaries. Precision is the ratio of correctly predicted boundaries over all predicted boundaries, recall is the ratio of correctly predicted boundaries over all true boundaries, and F-measure is the harmonic mean of precision and recall. The other set was standardly used in text segmentation literature: P_k [Beeferman et al., 1999] and WindowDiff (*WD*) [Pevzner and Hearst, 2002]. P_k is the probability of segmentation error that evaluates whether two lines l_i and l_j in lyrics fewer than k lines apart are incorrectly concatenated or divided by a segmentation model. P_k is considered a more suitable measure than F-measure in text segmentation because it assigns partial credit to nearly correct estimations. *WD* is a variant of P_k that resolves a problem of P_k by penalizing false positives. We set the window size k of P_k and *WD* to one-half the average line length of the correct segments for each song in the test set.

3.4.2 Contributions of Different Features

We investigated the contribution of each feature set by conducting ablation tests over different combinations of feature sets. The results are shown in Table 3.2. **Random** denotes our baseline, a model selecting boundaries with uniform probability $P = 0.186$, the true frequency of boundaries ($P = 517,234/2,788,079$). RPF$*$ and TF$*$

Table 3.2: Results of ablation tests

Method	P_k (%)	WD (%)	Precision (%)	Recall (%)	F-measure (%)
Random	49.35	53.67	14.29	12.50	13.33
TF$*$	40.51	44.65	34.95	31.66	33.22
RPF$*$	**27.00**	**32.16**	56.05	59.42	57.68
Proposed (ALL)	27.22	32.22	56.58	60.65	**58.55**
Ablation test					
−RPF1	31.38	35.89	51.95	51.32	51.63
−RPF2	30.62	36.73	49.22	57.64	53.10
−RPF3	27.46	32.71	55.59	59.40	57.43
−RPF4	27.64	32.68	55.73	59.94	57.76
−TF1_Uni_gram	27.00	31.95	56.90	60.73	58.75
−TF1_Bi_gram	26.84	31.88	56.96	61.41	59.10
−TF1_Tri_gram	27.32	32.53	55.88	61.42	58.52
−TF2_Uni_gram	28.24	31.84	59.40	51.91	55.40
−TF2_Bi_gram	26.89	31.25	58.86	58.12	58.49
−TF2_Tri_gram	26.67	31.40	57.91	60.23	59.05
−TF1_{Uni,Bi}_gram, TF2_Tri_gram (Best Performance)	**26.58**	31.55	57.40	61.21	**59.24**

denote the models with all repeated pattern features and all textual features, respectively. **Proposed** indicates the performance of the model with all proposed features. At the top of Table 3.2, the F-measure of the proposed method was 58.44, or 45 points higher than that of the random baseline.

The results of the ablation tests are shown in the bottom of Table 3.2. For example, "−RPF1" indicates that we ablated the feature RPF1 from the proposed method, which uses all of the features. Our best-performing model achieved an F-measure of 59.24 by excluding the TF1 unigram and bigram and TF2 trigram features.

The table shows that each type of our RPF features contributes to performance. Note that these four types are not redundant, and each of our hypotheses yielded positive results. Note that removing RPF1 and RPF2, which are intended to capture long-range repeated patterns, decreased the F-measure by 6.92 and 5.45 points, respectively. This result supports the hypothesis that sequences of repeated lines (diagonal lines in the SSM) are important clues for modeling lyrics segmentation.

In contrast to results reported in text segmentation literature [Beeferman et al., 1999], TF features turned out to be ineffective for lyrics segment boundary estimation,

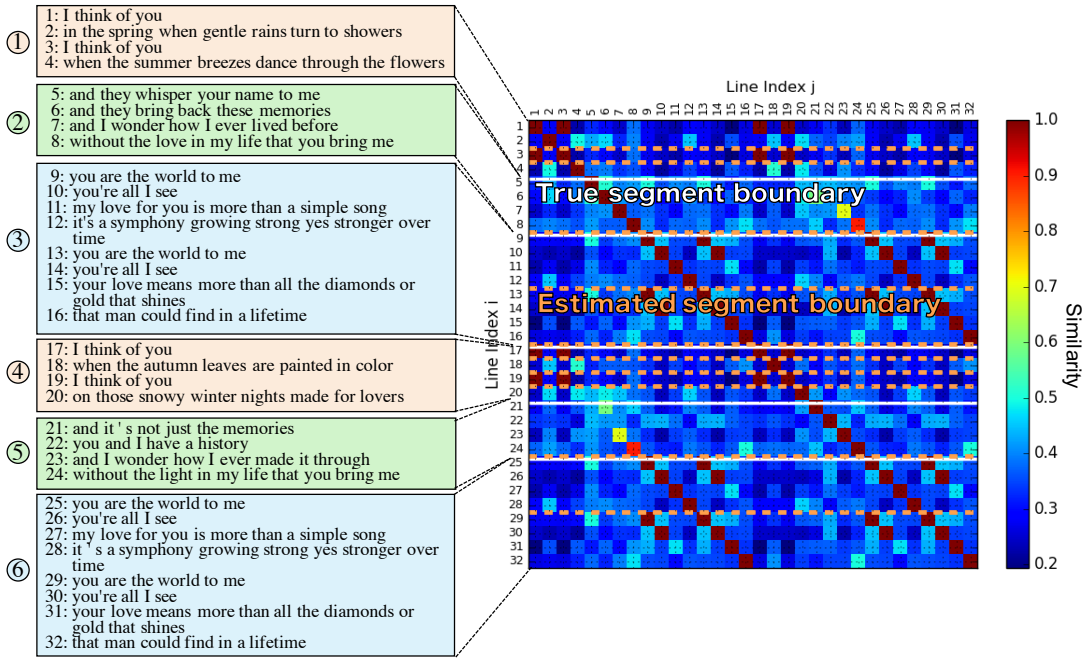


Figure 3.5: Examples of false positives (title of lyrics: *I think of you* (RWC-MDB-P-2001 No.87 from RWC Music Database [Goto et al., 2002])). White horizontal lines indicate true segment boundaries. Orange horizontal dashed lines indicate predicted boundaries.

except for TF2 unigram features. One possible reason is that there is a larger variety of expressions used at the beginning or end of a segment in lyrics compared with prose texts. Still, the inclusion of some textual features did lift the performance of the RPF* model by nearly 2 points. Further investigation of TF features is left for our future work.

3.4.3 Error Analysis

Figures 3.5 and 3.6 give two examples of lyrics and SSMs that illustrate typical errors of our best model. Horizontal dashed lines depict predicted boundaries. As shown in Figure 3.5, the model sometimes overly divides a true segment into segments as small as single lines, false positives that appear to be due to occurrences of repeated single lines (here, lines 1, 3, 17 and 19). This is not a trivial problem because repetitions of single lines sometimes serve as an important clue. In fact, when restricting diagonal

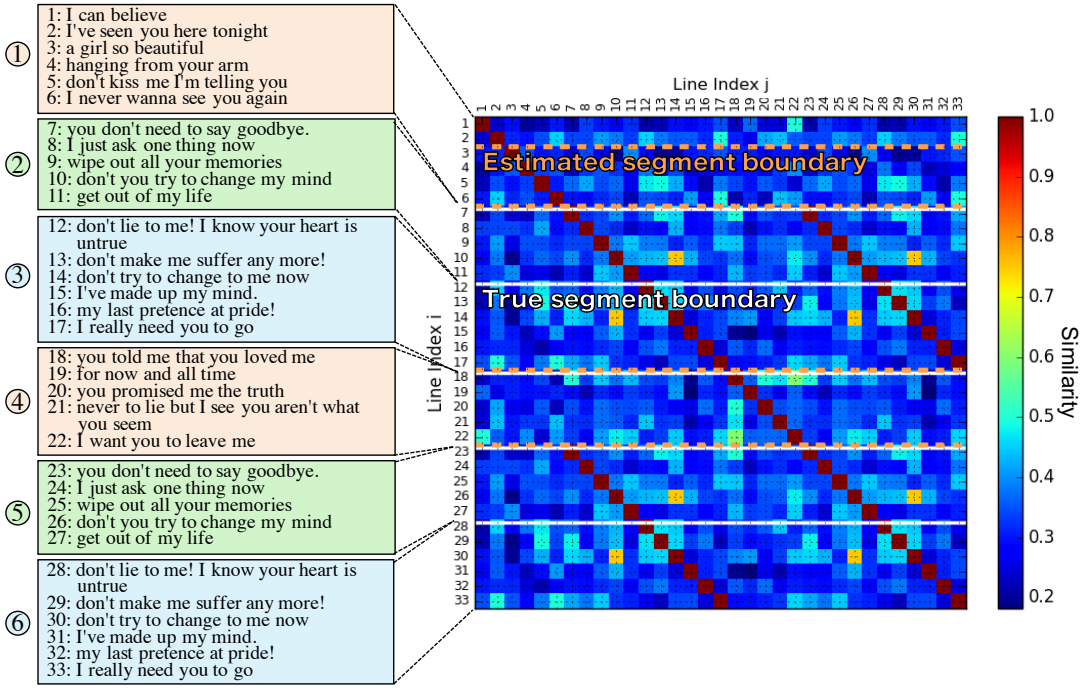


Figure 3.6: Examples of false negatives (title of lyrics: *Don't lie to me* (RWC-MDB-P-2001 No.97 from RWC Music Database [Goto et al., 2002])). White horizontal lines indicate true segment boundaries. Orange horizontal dashed lines indicate predicted boundaries.

lines to be of the length of two or more lines, we considerably lose recall while gaining precision. More investigation is needed for further improvement.

In contrast to the case of Figure 3.5, Figure 3.6 shows a typical example of false negatives. We missed a boundary between, for example, lines 11 and 12. For this boundary, we cannot find any clear repeated pattern indicator. Such cases suggest a limitation of repeated pattern features and the need for further refinement of the model. One direction is to incorporate semantics-oriented state-of-the-art techniques for prose text segmentation such as topic tiling [Riedl and Biemann, 2012].

3.5 Conclusion

This chapter has addressed the issue of modeling discourse segments in lyrics in order to understand and model the discourse-related nature of lyrics. We first conducted a large-scale corpus study into the discourse segments of lyrics, in which we examined our primary hypothesis that discourse segmentations strongly correlate with repeated patterns. To the best of our knowledge, this is the first study that takes a data-driven approach to explore the discourse structure of lyrics in relation to repeated patterns. We then proposed a task to automatically identify segment boundaries in lyrics and explored machine learning-based models for the task with repeated pattern features and textual features. The results of our empirical experiments show the importance of capturing repeated patterns in predicting the boundaries of discourse segments in lyrics. We need to refine the model further by incorporating topic/semantic information, to extend the modeling of lyric discourse by combining it with audio musical structure, and to embed a resulting model into application systems, such as lyrics generation systems and lyrics composition support systems.

Chapter 4

Modeling Storylines

In the previous chapter, we model the discourse structure of lyrics from the viewpoint of repeated patterns. However, using repeated patterns is insufficient to model discourse structures and it is necessary to capture the semantics of lyrics such as storylines between segments. This chapter focus on two notions which characterize lyrical discourse: *storyline* and *theme*. Both notions are described in textbooks on lyrics writing [Austin et al., 2010; Ueda, 2010].

A segment of lyrics is assumed to have its own purpose, which corresponds to a discourse segment purpose in terms of discourse analysis research [Grosz and Sidner, 1986]. In Figure 1.2, for example, Segment 6 introduces the story, Segment 7 retrospects a past event, and Segments 8 and 9 express an emotion which arises from the retrospection. We model a storyline as such a chain of coherent shifts between discourse segment purposes. Specifically, we capture typical types of discourse segment purposes as *latent topics* by applying topic modeling techniques [Blei et al., 2003] to a large collection of lyrical texts, and then model typical storylines of lyrics as a probability distribution over the transition of latent topics over successive segments (Figure 4.1). On top of storylines, we additionally consider the notion of theme, which we assume to be an entire discourse purpose. We assume that each song has at least one theme and each theme affects the distribution over both topic transitions and word choices. For the lyrics in Figure 1.2, for example, our model provides a result with which we can understand its theme as "Sweet Love" and estimates the *theme-sensitive* distributions over topic transitions and word choices.

In order to examine how well our model of lyrics fit real-world data, we experi-

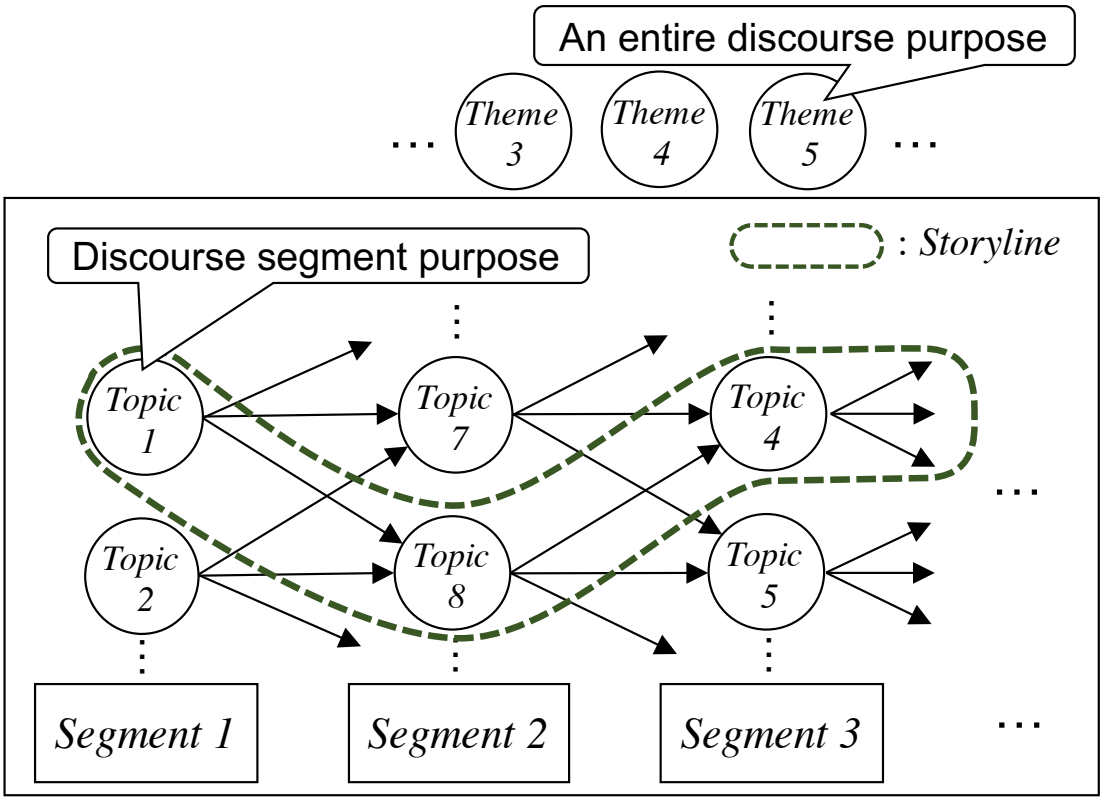


Figure 4.1: The notion of storylines and themes in lyrics.

ment with two distinct prediction tasks, word prediction and segment order prediction, and compare four variant models with different settings for considering storylines and themes. In the experiments, the models were trained with unsupervised learning over a large-scale corpus of popular music lyrics for both English and Japanese (around 100 thousand songs). The results demonstrate that the consideration of storylines (topic transitions) and themes contributes to improved prediction performance.

4.1 Overview of Topic Sequence Model

Transition of topics has been intensively studied in the context of topic modeling for sequential text data such as newspapers, weblogs, and conversations. Iwata et al. [2009] proposed a Topic Tracking Model (TTM), an extension of LDA, to models topic sequences. In the TTM, the topic distribution changes at each time. The TTM analyzes

changes in user interest (e.g., interest in weblogs and microblogs). Blei and Lafferty [2006] proposed the Dynamic Topic Model (DTM), which is similar to the TTM. In the DTM, the prior distribution of topic distribution changes at each time. The DTM analyzes changes in topic over time (e.g., topics in news articles and academic papers). The TTM and DTM have a topic distribution for a specific date (e.g., the DTM can train the topic distribution in a given period). Although the DTM and TTM can represent the topic sequence, extending these model to lyrics is difficult because, in lyrics, a segment's topic is time-independent.

Barzilay and Lee [2004] proposed Content Model (CM), which is typically used for discourse analysis, to model topic sequences in documents without date information. CMs are sentence-level hidden Markov models that capture the sequential topic structure of a news event, such as earthquakes. Several studies extended Barzilay's model to dialog acts (e.g., questions and responses) [Ritter et al., 2010; Zhai and Williams, 2014]. Ritter et al. [2010] assumed that an observed sentence is generated from either a dialog act-specific language model (e.g., questions and responses) or a dialog-specific language model (e.g., food and music). Zhai and Williams [2014] assumed that an observed word is generated from either a CM or an LDA and modeled the latent structure in task-oriented dialog. In their study, the sequential structure of dialog is modeled as a transition distribution.

We share the core concept as these studies and apply a CM to lyrics to model storylines (See Section 4.2.2). We then extend the CM to capture theme and investigate the effects of considering themes on top of storylines in our experiments.

4.2 Model Construction

Our final objective is to model the storyline of lyrics. However, precise modeling and representation of storylines remain an open issue. As mentioned previously, lyricists consider the order of topics when creating storylines; if the order changes, the content of the lyrics also changes. Therefore, we assume that a better storyline model can be used to predict the order of segments and the words in lyrics.

Based on the above assumption, we explore different topic sequence models to improve prediction performance. Lyricists often consider the order of topics when they create storylines; therefore, we assume that topic sequences can be represented as

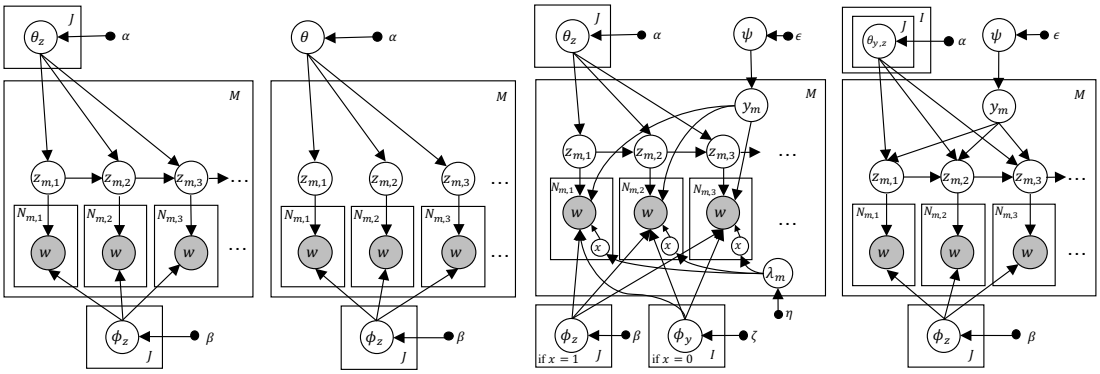


Figure 4.2: Plate notation of base models (a, b) and the proposed combination models (c, d). The shaded nodes are observed variables, dots are hyperparameters, x, y, and z are latent variables, and ψ, θ, ϕ, and λ are probability distributions.

a probabilistic distribution of transitions over latent states. Since lyricists often assign a certain role to each segment, we assume that the segment is in one latent variable for a given lyrical content and words are derived from each latent state. Moreover, we assume that lyrics in a song are in one latent variable (i.e., a theme) because lyricists often create storylines according to themes. Based on the above idea, we prepared four data-driven Bayesian models. By comparing the performance of the models, we investigate which encoding method can better model the storyline.

In the following, we first describe the notations used in this study and two baseline methods for modeling the storyline. Finally, we propose two extended combination models to handle theme and storyline simultaneously.

4.2.1 Preliminaries

We assume that we have a set of M lyrics (songs). The lyric is an index between 1 and M, where M is the number of songs. The m-th lyric contains S_m segments and has a single theme denoted as the latent variable y_m. The theme is an index between 1 and I, where I is the number of themes. The s-th segment contains a bag of words denoted as $\{w_{m,s,1}, w_{m,s,2}, ..., w_{m,s,N_{m,s}}\}$, where $w_{m,s,n}$ is an index between 1 and V, where V is the vocabulary size. $N_{m,s}$ is the number of words in the s-th segment of the m-th lyric. In addition, each segment has a single topic denoted as the latent variable $z_{m,s}$. The

topic is an index between 1 and J, where J is the number of topics. The storyline is represented as the sequence of a segment's topic denoted as $\mathbf{z}_m = z_{m,1}, z_{m,2}, ..., z_{m,S_m}$.

4.2.2 Base Model 1: Content Model

We use the CM Barzilay and Lee [2004] as a baseline model for the storyline of lyrics because this model is the simplest topic transition model that satisfies our assumption that *the topic sequence can be encoded as probabilistic latent state transition*. As shown in Figure 5.6(a), we assume that the storyline can be generated from a topic transition distribution θ_z. For the s-th segment in the m-th lyric, each topic $z_{m,s}$ is generated from the previous topic $z_{m,s-1}$ via the transition probability $P(z_{m,s}|\theta_{z_{m,s-1}})$. This probability is calculated by J-dimensional multinomial distribution θ_z drawn from Dirichlet distributions with symmetric hyperparameter α. Then the word $w_{m,s,n}$ in each segment is generated from each topic $z_{m,s}$ via topic-specific generative probability $P(w_{m,s,n}|\phi_{z_{m,s}})$. This probability is calculated by V-dimensional multinomial distribution ϕ_z drawn from Dirichlet distributions with symmetric hyperparameter β.

4.2.3 Base Model 2: Mixture of Segment Model

To investigate the effects of capturing topic transitions, we also build a model that removes topic transitions from the CM (Figure 5.6(b)). We refer to this model as the Mixture of Segment Model (MSM). In the MSM, each segment's topic $z_{m,s}$ is generated via the probability without transition $P(z_{m,s}|\theta)$. This probability is calculated by J-dimensional multinomial distribution θ drawn from Dirichlet distributions with symmetric hyperparameter α.

4.2.4 Proposed Model 1: Mixture of Unigram and Content Model

To verify that both theme and topic transition are useful for modeling the storyline, we propose a model that combines the theme and the topic transition simultaneously, and we compare this model to the baseline models. The idea behind this combined modeling is that we can mix a theme-specific model and the topic transition model (i.e., the CM) using linear interpolation assuming that words in lyrics are dependent on both the theme and the topic.

We use the Mixture of Unigram Model (MUM) [Nigam et al., 2000] as the theme-specific model because it is the simplest model that satisfies our assumption; *lyrics in a song are in a single latent variables (i.e., the theme)*. The MUM assumes that theme y_m is drawn from an I-dimensional theme distribution ψ and all words in the lyrics are drawn from V-dimensional multinomial distribution ϕ_{y_m} as shown in Figure 5.6(c).

In the proposed MUM-CM, we define a binary variable $x_{m,s,n}$ that uses either the MUM or the CM when the word $w_{m,s,n}$ is generated. Here, if $x_{m,s,n} = 0$, the word is drawn from the MUM's word distribution ϕ_y, and if $x_{m,s,n} = 1$, the word is drawn from the CM's word distribution ϕ_z. The binary variable x is drawn from a Bernoulli distribution λ_m drawn from a beta distribution with symmetric hyperparameter η. In other words, the words depend on both theme and topic, and the MUM and CM are defined independently in this model.

Figure 5.6(c) shows the plate notation of the MUM-CM. The generation process in the MUM-CM is as follows.

1. Draw a theme distribution $\psi \sim Dir(\epsilon)$
2. For each theme $y = 1, 2, ..., I$:
   - Draw a distribution of theme words $\phi_y \sim Dir(\zeta)$
3. For each topic $z = 1, 2, ..., J$:
   - Draw a topic transition distribution $\theta_z \sim Dir(\alpha)$
   - Draw a distribution of topic words $\phi_z \sim Dir(\beta)$
4. For each lyric $m = 1, 2, ..., M$:
   - Draw a theme $y_m \sim Multi(\psi)$
   - Draw a distribution of binary variable $\lambda_m \sim Beta(\eta)$
   - For each segment $s = 1, 2, ..., S_m$:
     - Draw a topic $z_{m,s} \sim Multi(\theta_{z_{m,s-1}})$
     - For the n-th word $w_{m,s,n}$ in segment s:
       - Draw a binary variable $x_{m,s,n} \sim Bernoulli(\lambda_m)$
       - If $x_{m,s,n} = 0$:
         - Draw a word $w_{m,s,n} \sim Multi(\phi_{y_m})$
       - If $x_{m,s,n} = 1$:
         - Draw a word $w_{m,s,n} \sim Multi(\phi_{z_{m,s}})$

Here, α, β, ϵ, and ζ are the symmetric hyperparameters of the Dirichlet distribution and η is the symmetric hyperparameter of the beta distribution. The generation probability of the m-th lyric is calculated as follows:

$$P(m) = P(x=0|\lambda_m) \sum_{y=1}^{I} \Big(P(y|\psi) \prod_{s=1}^{S_m} \prod_{n=1}^{N_{m,s}} P(w_{m,s,n}|\phi_y) \Big)$$
$$+P(x=1|\lambda_m) \sum_{\mathbf{z}_{all}} \prod_{s=1}^{S_m} \Big(P(z_s|\theta_{z_{s-1}}) \prod_{n=1}^{N_{m,s}} P(w_{m,s,n}|\phi_{z_s}) \Big) \tag{4.1}$$

where $\mathbf{z}_{all}$ denotes all possible topic sequences. If $s = 1$, θ_{z_0} denotes the initial state probabilities. This equation represents that a word $w_{m,s,n}$ is generated from the MUM according to $P(x=0|\lambda_m)$ or is generated from the CM according to $P(x=1|\lambda_m)$.

We use collapsed Gibbs sampling for model inference in the MUM-CM. For a lyric m, we present the conditional probability of theme y_m for sampling:

$$P(y_m = i|\mathbf{y}_{\neg m}, \mathbf{w}, \epsilon, \zeta) \propto P(y_m = i|\mathbf{y}_{\neg m}, \epsilon) \cdot P(\mathbf{w}_m|\mathbf{w}_{\neg m}, y_m = i, \mathbf{y}_{\neg m}, \zeta) \tag{4.2}$$

where $\mathbf{y}_{\neg m}$ denotes the topic set except the m-th lyric, $\mathbf{w}$ denotes the word set in the training corpus, $\mathbf{w}_m$ denotes the word set in the m-th lyric, and $\mathbf{w}_{\neg m}$ denotes the word set in the training corpus except $\mathbf{w}_m$.

We sample topic $z_{m,s}$ for a segment s of lyric m according to the following transition distribution:

$$P(z_{m,s} = j|\mathbf{z}_{\neg(m,s)}, \mathbf{w}, \alpha, \beta) \propto P(z_{m,s} = j|\mathbf{z}_{\neg(m,s)}, \alpha)$$
$$\cdot P(\mathbf{w}_{m,s}|\mathbf{w}_{\neg(m,s)}, z_{m,s} = j, \mathbf{z}_{\neg(m,s)}, \beta) \tag{4.3}$$

where $\mathbf{z}_{\neg(m,s)}$ denotes the topic set except the s-th segment in the m-th lyric, $\mathbf{w}_{m,s}$ denotes the word set in the s-th segment of the m-th lyric, and $\mathbf{w}_{\neg(m,s)}$ denotes the word set in the training corpus except $\mathbf{w}_{m,s}$.

For the n-th word in the segment s in the m-th lyric, we present the conditional

Algorithm 1 Model inference for the MUM-CM

```
Initialize parameters in the MUM-CM
for each iteration do
    for each lyrics m in the corpus do
        sample y_m according to Eq. 4.2
        for each segment s in m do
            sample z_{m,s} according to Eq. 4.3
            for each word w in s do
                sample x according to Eq. 4.4
            end for
        end for
    end for
    update hyperparameters by using fixed point iteration
end for
```

probability of its binary variables $x_{m,s,n}$ for sampling:

$$P(x_{m,s,n} = k|\mathbf{x}_{\neg(m,s,n)}, \mathbf{w}, \eta, \zeta, \beta) \propto P(x_{m,s,n} = k|\mathbf{x}_{\neg(m,s,n)}, \eta) \cdot P(w_{m,s,n}|\mathbf{w}_{\neg(m,s,n)}, x_{m,s,n} = k, \mathbf{x}_{\neg(m,s,n)}, \zeta, \beta) \tag{4.4}$$

where $\mathbf{x}_{\neg(m,s,n)}$ denotes the binary variable set except $x_{m,s,n}$. Note that the value of k is always 0 or 1.

We estimate hyperparameters α, β, ϵ, ζ, and η using fixed point iteration [Minka, 2000]. For each sampling iteration, the latent variables x, y, and z are sampled. Then, new hyperparameters are estimated such that the joint probabilities $P(\mathbf{w}, \mathbf{y}|\epsilon, \zeta)$, $P(\mathbf{w}, \mathbf{z}|\alpha, \beta)$, and $P(\mathbf{w}, \mathbf{x}|\eta)$ are maximized, where **y**, **z**, and **x** denote the latent variable sets in the training corpus.

In summary, the model and parameter inference for the MUM-CM is shown in Algorithm 1, and the update equations for Gibbs sampling are given in A.1.

4.2.5 Proposed Model 2: Mixture of Content Model

In the MUM-CM, we assume that theme and storyline are generated independently. On the other hand, as mentioned in the beginning of this chapter, lyricists often create storylines according to themes. Therefore, here, we propose the Mixture of Content Model (MCM) to verify this intuition. In the MCM, when a theme y is generated, a

storyline is generated using the theme-specific topic transition distribution $\theta_{y,z}$.

Figure 5.6(d) shows the plate notation of the MCM. The MCM generation process is as follows.

1. Draw a theme distribution $\psi \sim Dir(\epsilon)$
2. For each topic $z = 1, 2, ..., J$:
   - Draw a word distribution $\phi_z \sim Dir(\beta)$
   - For each theme $y = 1, 2, ..., I$:
     - Draw a topic distribution $\theta_{y,z} \sim Dir(\alpha)$
3. For each lyric $m = 1, 2, ..., M$:
   - Draw a theme $y_m \sim Multi(\psi)$
   - For each segment $s = 1, 2, ..., S_m$:
     - Draw a topic $z_{m,s} \sim Multi(\theta_{y_m, z_{m,s-1}})$
     - For n-th word $w_{m,s,n}$ in segment s:
       - Draw a word $w_{m,s,n} \sim Multi(\phi_{z_{m,s}})$

The generation probability of lyric m is calculated as follows:

$$P(m) = \sum_{y=1}^{I} \left(P(y|\psi) \sum_{\mathbf{z}_{all}} \prod_{s=1}^{S_m} \left(P(z_s|\theta_{y,z_{s-1}}) \prod_{n=1}^{N_{m,s,n}} P(w_{m,s,n}|\phi_{z_s}) \right) \right) \tag{4.5}$$

where $\mathbf{z}_{all}$ denotes all possible topic sequences. If $z = 1$, θ_{y,z_0} denotes the initial state probabilities. In this model, $P(y|\psi)$ represents the mixture ratio of the CMs.

We use collapsed Gibbs sampling for model inference in the MCM. For the m-th lyric, we present the conditional probability of theme y_m for sampling:

$$P(y_m = i|\mathbf{y}_{\neg m}, \mathbf{z}, \alpha, \epsilon) \propto P(y_m = i|\mathbf{y}_{\neg m}, \epsilon) \cdot P(\mathbf{z}_m|\mathbf{z}_{\neg m}, y_m = i, \mathbf{y}_{\neg m}, \alpha) \tag{4.6}$$

where $\mathbf{z}$ denotes the topic set in the training corpus, $\mathbf{z}_m$ denotes the topic sequence of lyric m (i.e., $z_{m,1}, z_{m,2}, ..., z_{m,S_m}$), and $\mathbf{z}_{\neg m}$ denotes the topic set in the training corpus except $\mathbf{z}_m$.

In the MCM, topic sequence $\mathbf{z}_m$ depends on theme y_m, as shown in Figure 5.6(d). Therefore, when a new theme y is sampled, the MCM must resample all topic sequences in lyric m simultaneously. To sample topic sequence $\mathbf{z}_m$, we present the following conditional probability:

$$P(\mathbf{z}_m|\mathbf{z}_{\neg m}, \mathbf{y}, \mathbf{w}, \alpha, \beta) \propto P(\mathbf{z}_m|\mathbf{z}_{\neg m}, \mathbf{y}, \alpha) \cdot P(\mathbf{w}_m|\mathbf{w}_{\neg m}, \mathbf{z}_m, \mathbf{z}_{\neg m}, \beta) \tag{4.7}$$

Algorithm 2 Model inference for the MCM

```
Initialize parameters in the MCM
for each iteration do
    for each lyrics m in the corpus do
        sample y_m according to Eq. 4.6
        sample z_m according to Eq. 4.7 by Forward Filtering-Backward Sampling
    end for
    update hyperparameters by using fixed point iteration
end for
```

However, enumerating all possible topic sequences is infeasible; thus, we use a Forward Filtering-Backward Sampling (FFBS) method [Scott, 2002] that can sample all latent states in a first-order Markov sequence using dynamic programming. In the FFBS method, the marginal probabilities of a topic sequence are calculated in the forward filtering step. Then, topics are sampled from the obtained probabilities in the backward sampling step. The hyperparameters α, β, and ϵ are estimated using fixed point iteration [Minka, 2000].

In summary, the model and parameter inference for the MCM is shown in Algorithm 2, and the update equations for Gibbs sampling are given in A.2.

4.3 Experiments

Here, we examine the effectiveness of the proposed models. First, we verify that topic transitions are useful for modeling storyline by evaluating the word prediction performance among different models. We then verify that the storyline correlates with the theme performing a segment order prediction task. Finally, we evaluate the proposed models qualitatively by exploring the trained topic transition diagrams.

In our experiments, we used two datasets that contain popular English and Japanese. In this study, we use an English lyrics database[1] and a corpus of Japanese lyrics collected from the Web. One issue that needed to be addressed prior to conducting the experiments was that no existing lyric corpora annotate verse-bridge-chorus tags. In this thesis, we assume that segment boundaries are indicated by empty lines inserted by lyricists. In addition, we assume that lyrics with storylines are divided into 6 to 18

[1]Music Lyrics Database. `http://www.odditysoftware.com/page-datasales1.htm`

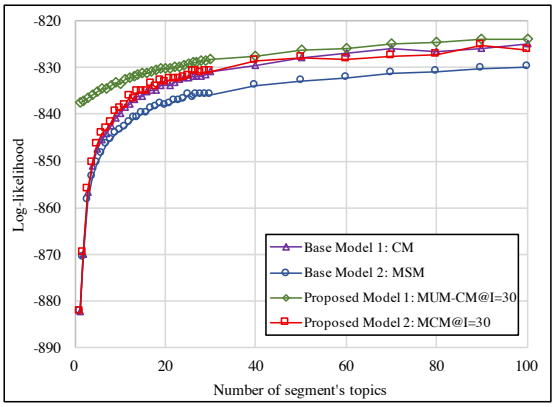


Figure 4.3: Log-likelihood on English test data under different segment topic settings (the number of themes I is fixed at 30).

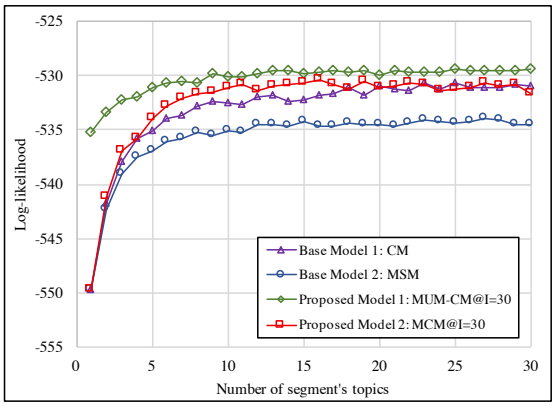


Figure 4.4: Log-likelihood on Japanese test data under different segment topic settings (the number of themes I is fixed at 30).

segments. The resulting dataset includes 80777 lyrics in the English dataset and 16563 lyrics in the Japanese dataset. We randomly split each collection into 60-20-20% divisions to construct the training, development, and test datasets.

We trained English-only and Japanese-only models. The collapsed Gibbs sampling ran for 1000 iterations, and the hyperparameters were updated for each Gibbs iteration. For training, we only used content words (nouns, verbs, and adjectives) because we assume that the theme and storyline can be represented using content words.

To extract content words, we use Stanford CoreNLP for English words [Manning et al., 2014] and the MeCab part-of-speech parser for Japanese words [Kudo et al., 2004].

4.3.1 Word Prediction Task

To verify that the topic transition and theme are useful properties for storyline modeling, we performed a word prediction task, which measures the test set generation probability. We assume that a better prediction model can capture the storyline of lyrics more effectively. In this experiment, we fixed the number of themes to 30 and computed the test set log-likelihood over the number of segment topics to compare different models.

Figure 4.3 and 4.4 show the English and Japanese test set log-likelihood under different segment topic settings. As can be seen, the CM outperforms the MSM, which

Table 4.1: Parameter tuning results with the development set.

Data	Model	I: # of themes	J: # of topics
English lyrics	Base Model 1: CM	none	30
	Proposed Model 1: MUM-CM	120	4
	Proposed Model 2: MCM	70	11
Japanese lyrics	Base Model 1: CM	none	15
	Proposed Model 1: MUM-CM	30	8
	Proposed Model 2: MCM	50	7

indicates that typical storylines were effectively captured as the probabilistic distribution of transition over latent topics of segments. Note that the proposed MUM-CM achieves the best performance, which indicates that a better storyline model can be constructed by assuming that the words in lyrics are generated from both theme and topic. The MCM, however, demonstrates only comparable performance to the CM despite that the MCM has a richer parameter space of topic transition distributions.

4.3.2 Segment Order Prediction Task

In this section, we verify that storyline correlates with theme. Here, we use the order test metric [Lapata, 2006], which is used to measure the predictive power of the sequential structure [Ritter et al., 2010; Zhai and Williams, 2014]. With the test order metric, the model predicts a reference segment order from all possible segment orders. However, enumerating all possible orders is infeasible; thus, we use the approximation method proposed by Zhai and Williams [2014]:

1. Select N permutations randomly from test data except reference order A.
2. Calculate the $N + 1$ document generative probabilities $P(m)$ whose order is A or N permutations.
3. Choose the hypothesis order A' whose generative probability is the best value in the $N + 1$ orders.
4. Compare the hypothesis order A' with the reference order A to calculate Kendall's tau:

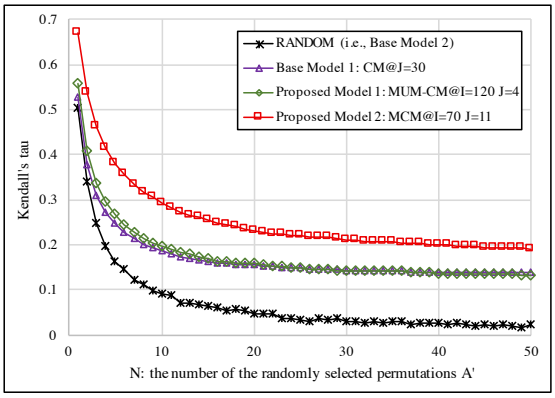


Figure 4.5: Average Kendall's τ for English lyrics against the number of random permutations.

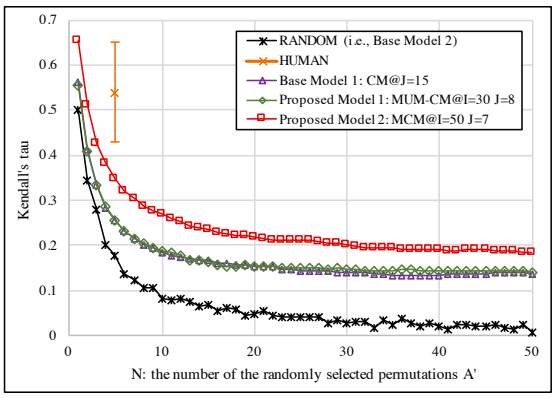


Figure 4.6: Average Kendall's τ for Japanese lyrics against the number of random permutations (the vertical range depicts the confidence intervals of the human assessment results).

$$\tau = \frac{c^+(A, A') - c^-(A, A')}{T(T-1)/2} \tag{4.8}$$

where $c^+(A, A')$ denotes the number of correct pairwise orders, $c^-(A, A')$ denotes the number of incorrect pairwise orders, and T denotes the number of segments in a lyric. Here, $N = 50$. This metric ranges from $+1$ to -1, where $+1$ indicates that the model selects the reference order and -1 indicates that model selects the reverse order. In other words, a higher value indicates that the sequential structure has been modeled successfully.

To tune the best parameters (i.e., the number of themes I and number of topics J), we use a grid search on the development set. Table 4.1 shows the parameters for each model that achieve the best segment order prediction task performance.

As a lower bound baseline, we use a model that randomly selects a hypothesis order A' (i.e., this lower bound is equivalent to the performance of Base Model 2 that does not handle topic transition). To obtain an upper bound for this task, nine Japanese evaluators selected the most plausible order from six orders that include a reference order. Here, $N = 5$ for the human assessments due to cognitive limitations relative to the number of orders. In this manual evaluation, each evaluator randomly selected unknown lyrics. As a result, we obtained 93 orders.

Figure 4.5 and 4.6 show Kendall's tau averaged over all English and Japanese

test data, respectively. The vertical range shows 95% confidence intervals for the human assessment results. The experimental results indicate that, compared to the lower bound, the proposed models that handle topic transition and theme (i.e., the MUM-CM and MCM) have the predictive power of the sequential structure. This result shows that topic transition and theme are useful properties for storyline modeling. The proposed MCM outperformed all other models on both test sets, while the MUM-CM only demonstrated performance comparable to that of the CM. We also conducted analysis of variance (ANOVA) followed by post-hoc Tukey tests to investigate the differences among these models ($p < 0.05$), drawing the conclusion that the difference between the MCM and the other models is statistically significant. These results show that storyline in lyrics correlates to theme. In contrast to the word prediction task, the MUM-CM has a similar predictive performance as the CM because the MUM-CM has only one topic transition distribution to model the order of segments, which is also the case for the CM.

For Japanese lyrics with $N = 5$, Figure 4.6 shows that Kendall's tau for the human evaluation was 0.58 ± 0.11, while the best performance of the model was 0.35. To investigate the cause of this difference, we asked the evaluators to write comments on this task. We found that most evaluators selected a single order by considering the following tendencies.

- Chorus segments tend to be the most representative, uplifting, and thematic segments. For example, the chorus often contains interlude words, such as "hey" and "yo", and frequently includes the lyrical message, such as "I love you". Moreover, the chorus is often the first or last segment; therefore, evaluators tend to first guess which segment is the chorus.

- Verse segments tend to repeat less frequently than choruses.

The human annotators were able to take these factors into account whereas the proposed models cannot consider verse-bridge-chorus structure. This issue could be addressed by combining the storyline of lyrics with the musical structure. We believe this direction will open an intriguing new field for future exploration.

Table 4.2: Representative words of each topic for English lyrics in MCM@$I = 70, J = 11$. The topic label indicates our arbitrary interpretation of the representative words.

z	Label	Representative words in each topic (top 40 words from $P(w\|\phi_s)$)
1	Abbreviation	ah, mi, dem, di, yuh, man, nah, nuh, gal, fus, work, inna, woman, pon, gim, fi, dat, seh, big, mek, weh, u, jump, wah, deh, yah, wid, tek, jah, waan, wine, red, !!!, youth, Babylon, ghetto, neva, hurry, l, nuff
2	Spanish	que, de, tu, el, te, lo, se, yo, un, e, si, por, con, como, amor, una, ti, le, quiero, para, sin, mas, esta, pa, pero, todo, al, solo, las, cuando, hay, voy, corazon, che, soy, je, los, del, vida, tengo
3	Exciting	like, hey, dance, uh, ya, right, body, party, put, shake, move, hand, hot, everybody, boy, beat, floor, c'mon, play, show, 'em, club, bang, drop, huh, lady, bounce, clap, sexy, freak, check, pop, push, low, top, shawty, boom, step, hip, dj
4	Religious	come, day, sing, god, song, lord, hear, Christmas, call, bring, child, new, heaven, beautiful, well, king, name, Jesus, pray, soul, angel, wish, yes, help, year, bear, happy, people, joy, old, son, Mary, bell, peace, father, mother, ring, holy, praise, voice
5	Love	love, feel, need, heart, hold, give, fall, night, dream, world, eye, light, tonight, shine, little, rain, fly, sun, touch, inside, fire, sky, kiss, free, sweet, star, cry, burn, true, close, mine, arm, alive, set, tear, somebody, open, higher, deep, blue
6	Explicit	nigga, shit, fuck, bitch, cause, money, niggaz, ass, hit, real, y', wit, hoe, game, street, em, bout, fuckin, gettin, rap, gun, blow, hood, kid, pay, damn, catch, block, tryin, aint, thug, motherfucker, dick, smoke, straight, house, g, talkin, dog, buy
7	Locomotion	go, get, let, back, ta, take, keep, home, round, turn, run, rock, ride, long, stop, roll, ready, got, road, high, slow, far, music, train, start, town, goin, please, drive, control, radio, fight, fast, car, city, ground, rollin, foot, comin, outta
8	Interlude	oh, la, yeah, ooh, da, whoa, ba, ha, doo, woah, yea, ay, ho, ohh, oooh, mmm, ooo, woo, hoo, oo, dum, ohhh, oh-oh, ahh, ooooh, oooo, wee, la., ohhhh, click, dee, fa, bop, shame, l.a., hmmm, ahhh, drip, trouble, mm
9	Feeling	know, say, time, never, see, make, one, way, think, life, thing, try, find, leave, look, nothing, always, everything, believe, change, lose, live, mind, much, something, wait, better, 'cause, break, wrong, lie, hard, end, word, stay, mean, seem, friend, someone, care
10	Love	na, wan, gon, baby, girl, want, tell, good, bad, alright, talk, crazy, nobody, cuz, im, ai, babe, bye, dont, lovin, fine, feelin, worry, pretty, phone, nothin, fun, thinkin, guy, cos, kind, spend, doin, next, number, sex, treat, cool, honey, cant
11	Life	head, walk, face, stand, watch, die, dead, black, sleep, blood, door, wake, line, wall, kill, water, wind, room, white, sit, hide, grow, bed, fear, lay, rise, hell, sea, meet, scream, pull, death, cut, window, begin, pass, fill, wear, skin, full

4.3.3 Analysis of Trained Topic Transitions

Our experimental results indicate that topic transition and theme are useful properties for modeling a storyline. Thus, we are interested in understanding what kinds of themes and topic transitions our model can acquire. Here, to interpret the proposed MCM, we examine word probabilities $P(w|\phi_z)$ and topic transition probability $P(s|\theta_{y,z'})$ and then visualize topic transition diagrams. To clarify our topic transition analysis, we manually assigned labels to each topic by observing the word list whose generative probability $P(w|\phi_z)$ is a large value. Table 4.2 and 4.3 show the assigned labels and representative words for the topics in the English and Japanese models, respectively. For each topic, we list the top 40 words in decreasing order of word probability $P(w|\phi_z)$. Figure 4.7 and 4.8 show the transition diagrams for some themes

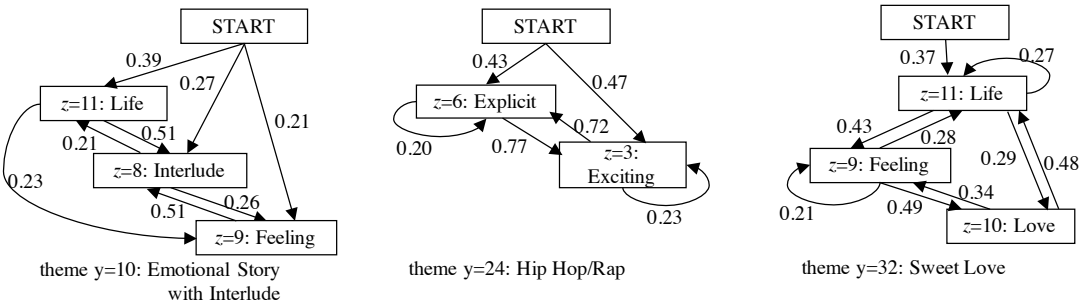


Figure 4.7: Examples of English MCM($I = 70$, $J = 11$) transitions between topics for each theme (see Table 4.2 for word lists). The theme label indicates our arbitrary interpretation of the topic transitions.

in the English and Japanese models, respectively. Here, each arrow indicates higher transition probabilities ($P(s|\theta_{y,z'}) > 0.20$), and each square node indicates the topic z. Note that the initial node ⟨START⟩ indicates the initial state $z = 0$.

We found the following reasonable storylines with the English model (Figure 4.7).

- In theme $y = 10$, we see the transition ⟨Life⟩ → ⟨Interlude⟩ → ⟨Feeling⟩. The topic ⟨Interlude⟩ comprises words such as *oh*, *la*, and *yeah* and acts as a bridge between the verse and the chorus.

- In theme $y = 24$, we see that the ⟨ Explicit ⟩ topic tends to shift to ⟨Exciting⟩, which contains words such as *dance*, *sexy*, and *pop*. This topic sequence appears frequently in hip hop/rap songs.

- In theme $y = 32$, we see the transition ⟨Life⟩ → ⟨Feeling⟩ → ⟨Love⟩. We arbitrarily decided the theme label of this topic transition as "Sweet Love". Here, the last topic ⟨Love⟩ tends to shift to the first topic ⟨Life⟩. This indicates that the model captures the repetition structure (e.g., *A-B-C-A-B-C*, where each letter represents a segment).

We also found the following reasonable storylines with the Japanese model (Fig.4.8).

- In theme $y = 6$, we observe the transition ⟨Scene⟩ → ⟨Lyrical⟩ → ⟨Love⟩, which is common in love songs.

- In theme $y = 12$, we see a transition among ⟨Life⟩, ⟨English⟩, and ⟨Exciting⟩, which often appears in Japanese hip hop/rap songs.

- In theme $y = 14$, we see a transition between $\langle$Clean$\rangle$ and $\langle$Lyrical$\rangle$, which is commonly seen in hopeful songs.

Although we selected these arbitrary diagrams to represent a reasonable storyline, in fact, the self-transition diagrams were trained using other themes. Note that the MCM learns different topic transition distributions according to different themes in an unsupervised manner. This shows that many lyricists consider the topic order and theme as described in textbooks [Austin et al., 2010; Ueda, 2010].

4.4 Conclusion

This chapter has addressed the issue of modeling the discourse nature of lyrics and presented the first study aiming at capturing the two common discourse-related notions: storylines and themes. We assumed that a storyline is a chain of transitions over topics of segments and a song has at least one entire theme. We then hypothesized that transitions over topics of lyric segments can be captured by a probabilistic topic model which incorporates a distribution over transitions of latent topics and that such a distribution of topic transitions is affected by the theme of lyrics.

Aiming to test those hypotheses, this study conducted experiments on the word prediction and segment order prediction tasks exploiting a large-scale corpus of popular music lyrics for both English and Japanese. The findings we gained from these experiments can be summarized into two respects. First, the models with topic transitions significantly outperformed the model without topic transitions in word prediction. This result indicates that typical storylines included in our lyrics datasets were effectively captured as a probabilistic distribution of transitions over latent topics of segments. Second, the model incorporating a latent theme variable on top of topic transitions outperformed the models without such variables in both word prediction and segment order prediction. From this result, we can conclude that considering the notion of theme does contribute to the modeling of storylines of lyrics.

This study has also shaped several future directions. First, we believe that our model can be naturally extended by incorporate more linguistically rich features such as tense/aspect, semantic classes of content words, sentiment polarity, etc. Second, it is also an intriguing direction to adopt recently developed word/phrase embeddings [Mikolov

Table 4.3: Representative words of each topic for Japanese lyrics in MCM@$I = 50, J = 7$. The topic label indicates our arbitrary interpretation of the representative words. English words are translated by the authors and original Japanese words are given in parentheses.

z	Label	Representative words in each topic: top 40 words from $P(w\|\phi_s)$
1	English	go, get, let, know, say, night, baby, time, good, way, feel, heart, take, day, dance, make, life, need, party, come, see, tell, dream, everybody, rock, stop, keep, happy, have, give, tonight, please, world, mind, hand, shake, rain, jump, try, your
2	Scene	town (*machi*), night (*yoru*), rain (*ame*), summer (*natsu*), come (*kuru*), window (*mado*), white (*shiroi*), snow (*yuki*), wait (*matsu*), room (*heya*), morning (*asa*), get back (*kaeru*), season (*kisetsu*), fall (*huru*), spring (*haru*), winter (*huyu*), blow (*huku*), wave (*nami*), cold (*tumetai*), hair (*kami*), shoulder (*kata*) , memory (*omoide*), back (*senaka*), run (*hashiru*), long (*nagai*), last (*saigo*), shadow (*kage*), sleep (*nemuru*), close (*tojiru*), finger (*yubi*), get wet (*nureru*), remember (*omoidasu*), quiet (*shizuka*), pass (*sugiru*), cheek (*ho*), fall (*otiru*), breath (*iki*), open (*akeru*), car (*kuruma*)
3	Exciting	go (*iku*), front (*mae*), no, sound (*oto*), dance (*odoru*), nothing (*nai*), fly (*tobu*), life (*jinsei*), can run (*hashireru*), begin (*hajimaru*), proceed (*susumu*), stand up (*tatu*), raise (*ageru*), freedom (*jiyu*), era (*jidai*), serious (*maji*), head (*atama*), body (*karada*), ahead (*saki*), power (*chikara*), throw (*suteru*), fire (*hi*), carry (*motu*), high (*hai*), take out (*dasu*), decide (*kimeru*), ride (*noru*), speed up (*tobasu*), Venus, Japan (*nihon*), maximum (*saikou*), rhythm (*rizumu*), non, up, rise (*agaru*), party (*patatii*), wall (*kabe*), companion (*nakama*), girl (*gaaru*), battle (*shobu*)
4	Love	love, love (*ai*), hug (*dakishimeru, daku*), kiss, feel (*kanjiru*), girl, pupil (*hitomi*), ardent (*atsui*), look on (*mitsumeru*), sweet, hold, lonely, sweet (*amai*), kiss (*kisu*), pair (*futari*), smile, stop (*tomeru*), miss, sorrowful (*setsunai*), moon, stop (*tomaru*), heart (*haato*), detach (*hanasu*), overflow (*afureru*), moment (*shunkan*), tempestuous (*hagesii*), moonlight, shine, lovin, touch (*fureru*), little, arm (*ude*), break (*kowareru*), angel (*tenshi*), beating (*kodo*), mystery (*fushgi*), destiny, miracle (*kiseki*), shinin
5	Clean	sky (*sora*), dream (*yume*), wind (*kaze*), light (*hikari*), flower (*hana*), star (*hoshi*), disappear (*kieru*), world (*sekai*), sea (*umi*), future (*mirai*), far (*toi*), voice (*koe*), moon (*tsuki*), shine (*kagayaku*), bloom (*saku*), flow (*nagareru*), sun (*taiyo*), place (*basho*), blue (*aoi*), reach (*todoku*), dark (*yami*), illuminate (*terasu*), cloud (*kumo*), destiny (*eien*), unstable (*yureru*), wing (*tsubasa*), deep (*fukai*), song (*uta*), continue (*tuduku*), sing (*utau*), pass over (*koeru*), shine (*hikaru*), look up (*miageru*), bird (*tori*), finish (*owaru*), color (*iro*), distance (*toku*), high (*takai*), rainbow (*niji*), be born (*umareru*)
6	Lyrical	now (*ima*), mind (*kokoro*), human (*hito*), heart (*mune*), believe (*shinjiru*), word (*kotoba*), oneself (*jibun*), live (*ikiru*), tear (*namida*), forget (*wasureru*), love (*aisuru*), know (*siru*), hand (*te*), cry (*naku*), tomorrow (*ashita*), walk (*aruku*), change (*kawaru*), strong (*tsuyoi*), feeling (*kimochi*), someday (*itsuka*), kind (*yasasii*), everything (*subete*), look (*mieru*), understand (*wakaru*), can be (*nareru*), smile (*egao*), happy (*siawase*), can do (*dekiru*), every day (*hibi*), outside (*soba*), crucial (*taisetsu*), road (*michi*), eye (*me*), look for (*sagasu*), convey (*tutaeru*), time (*jikan*), take leave (*hanareru*), guard (*mamoru*), be able to say (*ieru*)
7	Life	good (*yoi*), say (*iu*), like (*suki*), love (*koi, daisuki*), woman (*onna*), look (*miru*), man (*otoko*), laugh (*warau*), do (*yaru*), today (*kyo*), think (*omou*), spirit (*ki*), face (*kao*), no good (*dame*), listen (*kiku*), phone (*denwa*), tonight (*konya*), friend (*tomodachi*), reach (*tuku*), daughter (*musume*), bad (*warui*), meet (*au*), go (*iku*), appear (*deru*), adult (*otona*), together (*issyo*), good (*umai*), consider (*kangaeru*), die (*sinu*), stop (*yameru*), everyday (*mainichi*), story (*hanashi*), talk (*hanasu*), cheerful (*genki*), drink (*nomu*), human (*ningen*), job (*shigoto*), early (*hayai*)

et al., 2013; Pennington et al., 2014] to capture the semantics of lyrical phrases in a further sophisticated manner. Third, verse-bridge-chorus structure of a song is also worth exploring. Our error analysis revealed that the human annotators seemed to be able to identify verse-bridge-chorus structures and use them to predict segment orders. Modeling such lyrics-specific global structure of discourse is an intriguing direction of our future work. Finally, it is also important to direct our attention toward the integration of linguistic discourse structure of lyrics and music structure of audio signals. In

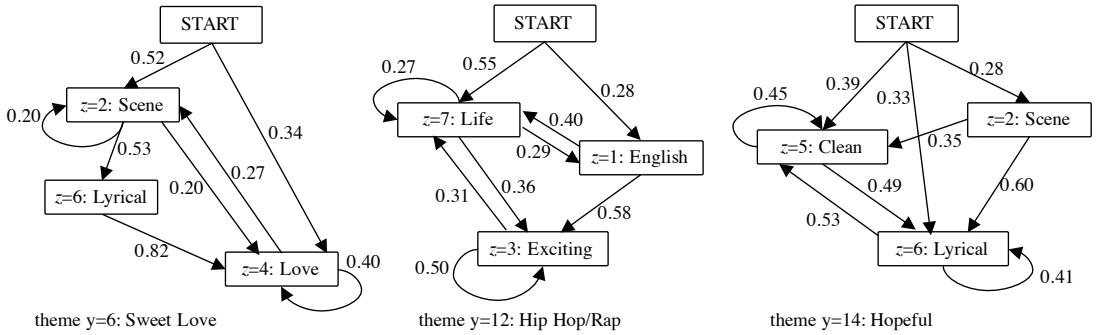


Figure 4.8: Examples of Japanese MCM($I = 50$, $J = 7$) transitions between topics for each theme (see Table 4.3 for word lists). The theme label indicates our arbitrary interpretation of the topic transitions.

this direction, we believe that recent advances in music structure analysis [Goto, 2006, etc.] can be an essential enabler.

Chapter 5

Modeling Relationship between Melody and Lyrics

In this chapter, We deeply analyze the correlation between melody and discourse structure of lyrics, and evaluate proposed model quantitatively, while prior exploration [Nichols et al., 2009] covers only correlations at the prosody level but not structural correlations of lyrics and melody. This direction of research, however, has never been promoted partly because it requires a large training dataset consisting of aligned pairs of lyrics and melody but so far no such data has been available for research. Therefore, we propose a methodology for creating melody-lyrics alignment data by leveraging lyrics and their corresponding musical score data on the web. We demonstrate that we can construct a relatively large-scale alignment data of 1,000 Japanese songs using this method.

Moreover, we propose novel lyrics generation models that generate lyrics for an entire input melody. We extend a common Recurrent Neural Network Language Model (RNNLM) Mikolov et al. [2010] so that its output can be conditioned on a featurized input melody. We also demonstrate how the efficiency of the model training can improve by training the model simultaneously for a mora count prediction subtask.

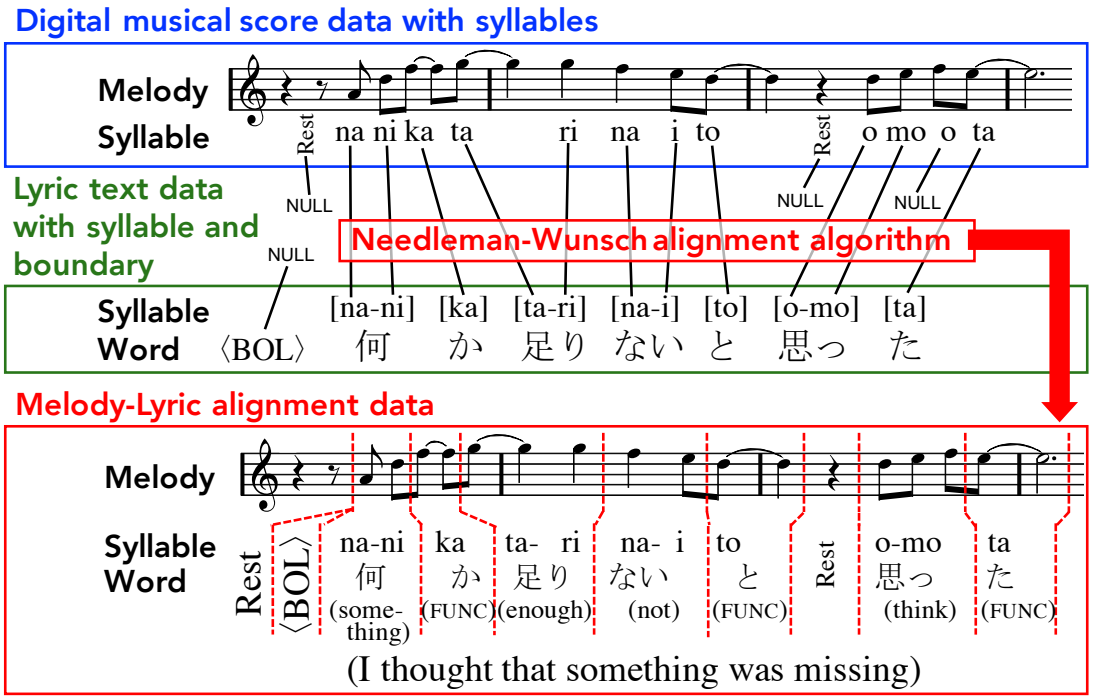


Figure 5.1: Automatic melody-lyric alignment using the Needleman-Wunsch algorithm. ⟨BOL⟩ indicates a line boundary.

5.1 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 in this study is a collection of melody-lyric pairs where the melody and lyrics are aligned at the level not only of note-mora alignment but also of linguistic components (i.e., word/sentence/paragraph boundaries) of lyrics, as illustrated in the bottom of Figure 5.1. We create such a dataset by automatically combining two types of data available from many forum sites: digital music score data for *vocal synthesizers* (the top of Figure 5.1) and raw lyric text data (the middle). A digital music score for a vocal synthesizer specifies a melody score augmented with mora information for each melody note (See the top of Figure 5.1). Recently, it is becoming increasingly popular for amateur music composers to upload their songs on Web forum sites, where visitors can freely play uploaded songs with a vocal synthesizer. Those forum sites can thus be considered as a useful, yet unexplored source of digital music score data that can

be used for research purposes. Score data augmented this way is sufficient for a vocal synthesizer to "sing" but is insufficient for our research goal. A lyrics is not just a sequence of moras but a meaningful sequence of words, which then further constitute a coherent sequence of sentences and paragraphs as discourse. This study aims for analyzing and modeling the correlations between patterns of melody and such linguistic structure of lyrics. For this purpose, we augment music score data further with boundaries of lyrics words, lines, and segments, where we assume that sentences and paragraphs of a lyrics are approximately captured by lines and segments,[1] respectively, of the lyrics in the raw text format.

The integration of digital music scores and raw lyric texts is achieved by (i) applying a morphological analyzer[2] to lyric texts for word segmentation and Chinese character pronunciation and (ii) aligning music score and lyric text at the moras level as illustrated in Figure 5.1. 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 in case where the morphological analysis fails in Chinese character pronunciation, which occurs for only less than 1% of given words.

With this procedure, we obtained 54,181 Japanese raw lyrics texts and 1,000 digital musical scores from online forum sites; we thus created 1,000 melody-lyrics pairs. In this data, ⟨BOL⟩ and ⟨BOS⟩ are special symbols denoting a line and a segment boundary, respectively. For selecting the 1,000 songs, we chose only songs with a high view count. We refer to these 1,000 melody-lyrics pairs as a *melody-lyrics alignment* data [3] and refer to the remaining 53,181 lyrics without melody as a *raw lyrics text* data.

We split 1,000 melody-lyrics alignments 900:100 into train and test sets. We use 53,181 raw lyrics texts as the train set. In those, we use 20,000 of the most frequent words whose mora counts are equal to or less than 10, and converted others to a special symbol ⟨unknown⟩. All the digital music score data we collected are 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

[1] We assume that segment boundaries are indicated by empty lines inserted.

[2] In order to extract word boundaries and mora information for Japanese lyrics, we apply MeCab part-of-speech parser [Kudo et al., 2004].

[3] Due to copyright protection for the music score and raw lyric text data, we cannot release our melody-lyric alignment data to the public. However, we will publicly release all source URLs (mostly taken from sites such as http://utaforum.net) of the 1,000 songs.

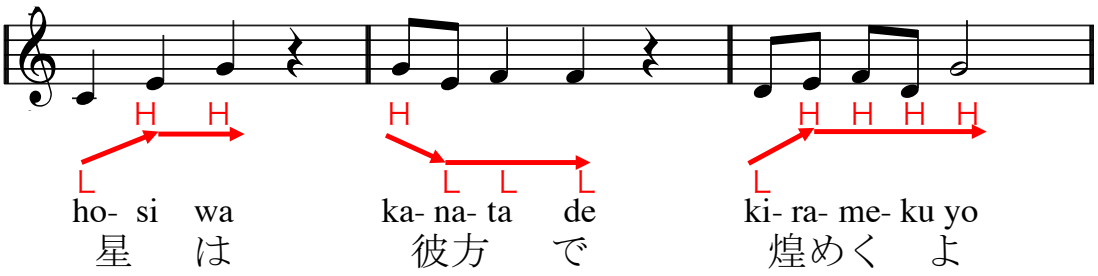


Figure 5.2: Melody and intonation. “H” indicates high intonation and “L” indicates low intonation.

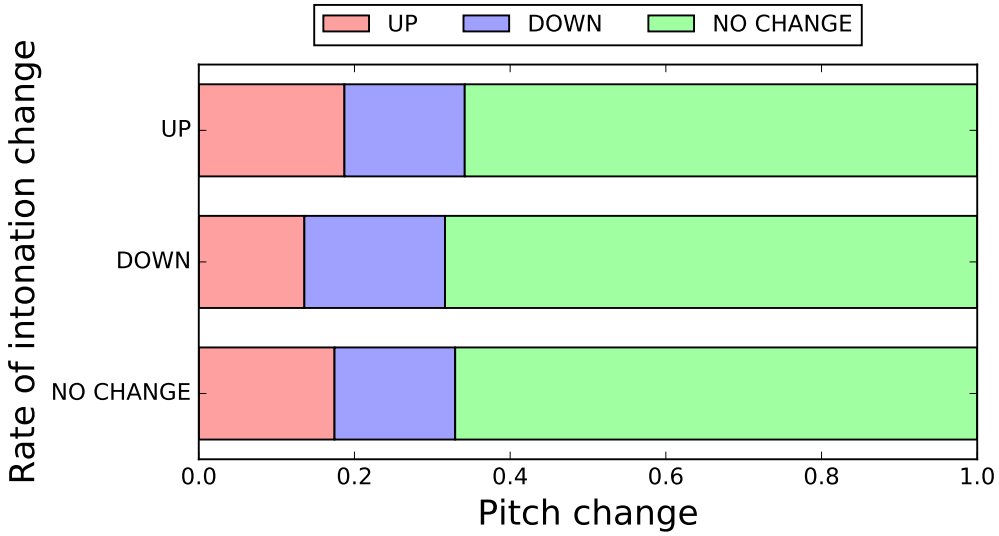


Figure 5.3: Relationship between pit changes and intonation changes.

method for data creation is general enough to be applied to other language formats such as MusicXML and ABC, because transferring a data format to UST is straightforward.

5.2 Correlations between Melody and Lyrics

In this section, we analyze the correlations between melody and lyrics using Melody-Lyrics alignment data of 1000 songs created in the previous section. First, we analyze the correlation between melodic pitch and intonation of lyrics and then analyze the correlation between notes/rests of melody and lines/segments of lyrics.

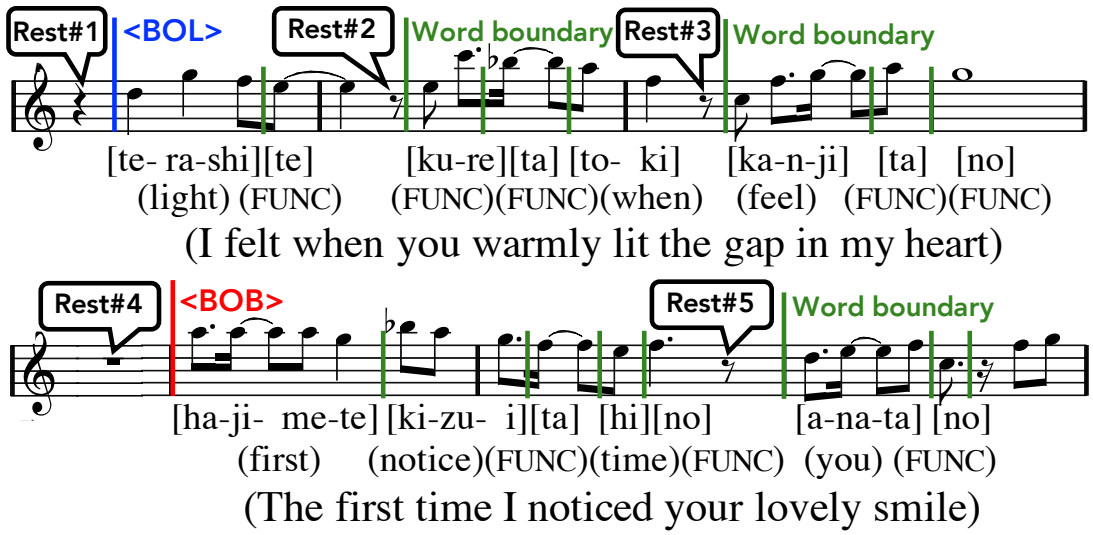


Figure 5.4: Example of boundaries appearing immediately after a rest. ⟨BOS⟩ indicates a segment boundary.

5.2.1 Melody and Intonation

In writing lyrics, it is said that the writer matches the intonation of the lyrics to the pitch of the melody [Ueda, 2010]. For example in Figure 5.2, the word's intonation also rises from "Low" to "High" when the pitch goes up. In this section, we divide the change of pitch and intonation into three states of "UP", "DOWN", and "NO CHANGE", and calculate the distribution of change of intonation when the pitch changes. Figure 5.3 shows that the intonation tends to rise when the pitch increases, while the intonation tends to fall when the pitch decreases. However, these distributions are only about 20%, and the difference between the rate of which intonation rise and the rate of which intonation fall is only about 5%. Here, we judge that there is no strong restriction on the correlation between pitch and intonation. Therefore, in this study, this correlation is ignored in the automatic lyrics generation task.

5.2.2 Melody and Line/Segment Boundaries

We examine two phenomena related to lyric boundaries: (1) the positions of lyric 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 segment boundary, as shown in Figure 5.4. All analyses were performed on the training split of the melody-lyric alignment data, which is described in the dataset section.

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 segment), we refer to the position of the beginning of the line (or segment).[1] In Figure 5.4, we say, for example, that the boundary of the first segment beginning "*te-ra-shi te*" coincides with Rest#1. The result, shown at the top of Figure 5.5, indicates that line and segment boundaries are strongly biased to rest positions and are far less likely to appear at note positions. Words, sentences and paragraphs rarely span beyond a long melody rest.

The bottom of Figure 5.5 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. Whereas this correlation may seem rather trivial, we would like to emphasize that no prior study provides any quantitative analysis of this phenomenon with this size of data. It is also important to note that the choice of segment boundaries looks like a probabilistic process (i.e., not all boundaries are determined by rests of a melody). 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 below.

5.3 Melody-Conditioned Generation of Lyrics

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 5.2. We propose to pursue this goal by conditioning a 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 5.6. Formally, we are given a melody $\mathbf{m} = m_1,...,m_i,...,m_I$ that is a sequence of notes and rests, where

[1] The beginning of the line/segment and the end of the line/segment are equivalent since there is no melody between the end and beginning of line/segment.

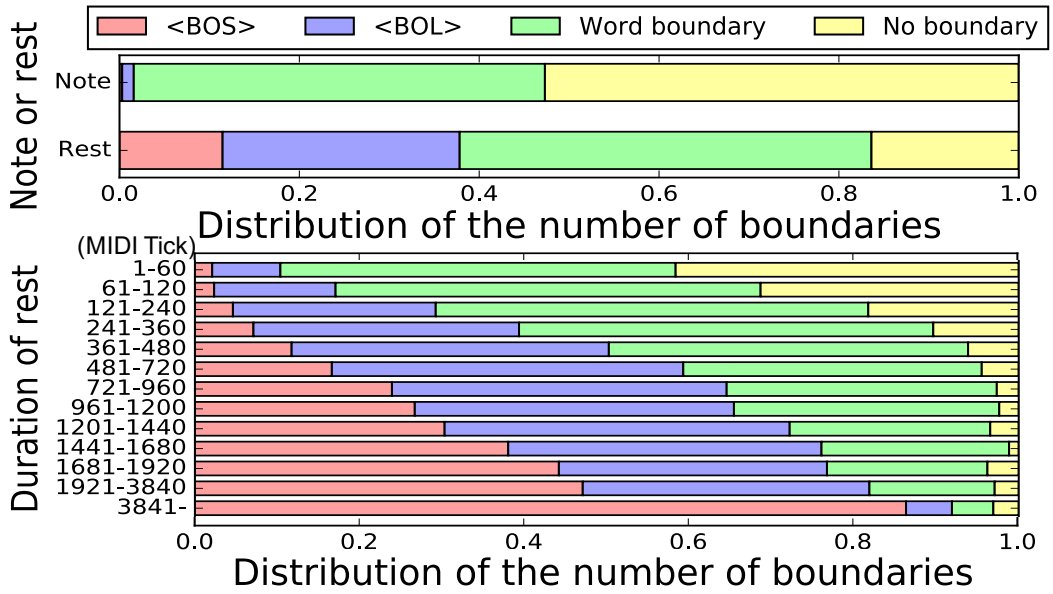


Figure 5.5: Distribution of the number of boundaries in the melody-lyric alignment data.

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: $\langle$BOL$\rangle$ and $\langle$BOS$\rangle$, special symbols denoting a line and a segment 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, as input. In this model, we assume that the moras 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 mora counts of the previously generated words.[1] 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 mora counts. As shown in Figure 5.5,

[1]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.

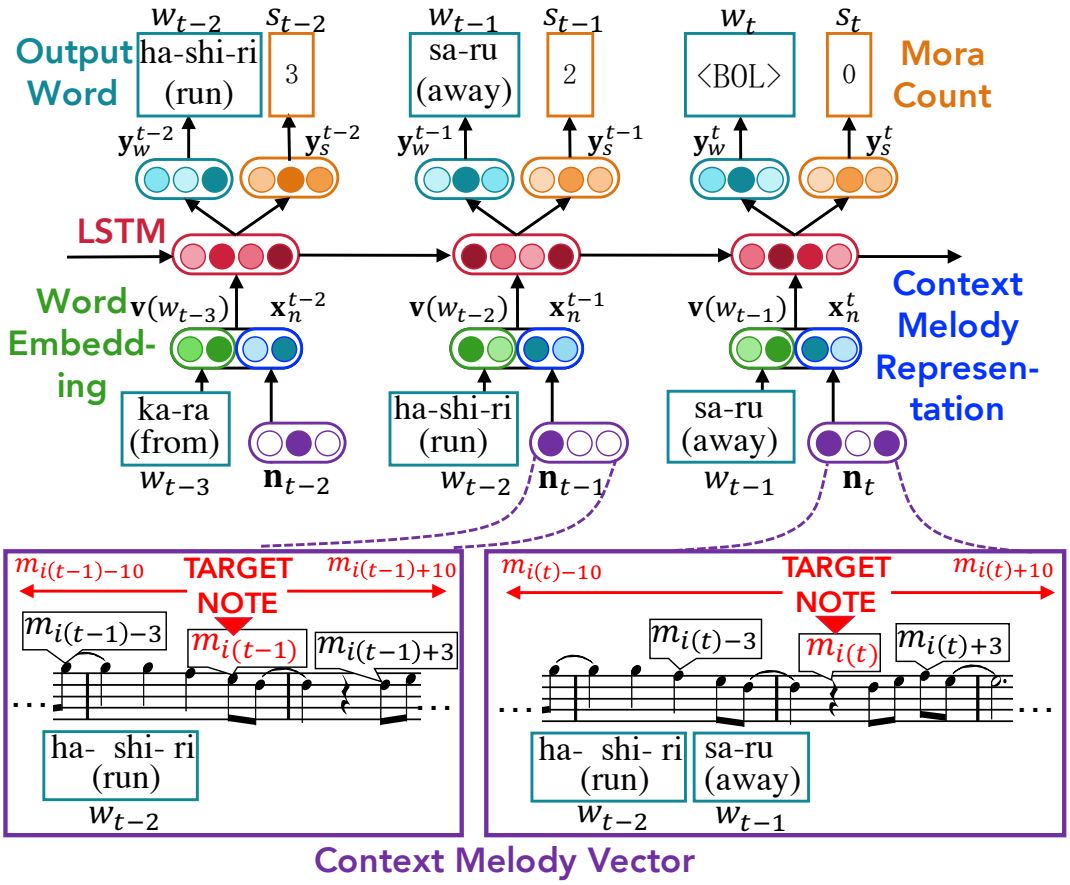


Figure 5.6: Melody-conditioned RNNLM.

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 5.6, it is desirable to select a word whose mora 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 mora count of a word. However, our melody-lyric 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 lyric texts that can be obtained in greater quantities. Second, we construct a model that

predicts the number of moras in each word, as well as words themselves, to explicitly supervise the correspondence between rest positions and mora 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-lyric alignment data.

5.3.1 Model construction

The proposed model is based on a standard RNNLM Mikolov et al. [2010]:

$$P(\mathbf{w}) = \prod_{t=1}^{T} P(w_t | w_0, ..., w_{t-1}), \tag{5.1}$$

where context words are encoded using LSTM [Hochreiter and Schmidhuber, 1997] and the probabilities over words are calculated by a $\mathrm{softmax}$ function. $w_0 = \langle \mathrm{B} \rangle$ is a symbol denoting a begin 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:

$$P(\mathbf{w}|\mathbf{m}) = \prod_{t=1}^{T} P(w_t | w_0, ..., w_{t-1}, \mathbf{n}_1, ..., \mathbf{n}_t). \tag{5.2}$$

The model simultaneously predicts the mora 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 mora counts:

$$P(\mathbf{s}|\mathbf{m}) = \prod_{t=1}^{T} P(s_t | w_0, ..., w_{t-1}, \mathbf{n}_1, ..., \mathbf{n}_t), \tag{5.3}$$

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

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

$$\mathbf{y}_w^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_w \mathbf{z}_t)), \tag{5.4}$$

$$\mathbf{y}_s^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_s \mathbf{z}_t)), \tag{5.5}$$

where $\mathbf{z}_t$ is the output of the LSTM for each time step. V is the vocabulary size and S is the mora count threshold.[1] $\mathbf{W}_w$ 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 $\mathbf{n}_t$:

$$\mathbf{x}^t = [\mathbf{v}(w_{t-1}), \mathbf{x}_n^t], \tag{5.6}$$

$$\mathbf{x}_n^t = \mathrm{ReLU}(\mathbf{W}_n \mathbf{n}_t + \mathbf{b}_n), \tag{5.7}$$

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. 5.2) using beam search. The model stops generating lyrics when the mora 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, while we experiment on Japanese lyrics data in this thesis. The model only requires the sequences of melody and words as input and does not use any language-specific features.

5.3.2 Context melody vector

In Section 5.2, we indicated that the positions of rests and their durations are important factors for modeling lyric boundaries. 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 5.6).

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[2], 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 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

[1] We exclude the words with mora count greater than 10 from the output vocabulary and replace these words with a symbol ⟨unknown⟩ in the training data. Additionally, we define the mora counts of the ⟨BOL⟩ and ⟨BOS⟩ as zero.

[2] We rounded each duration to one of the values 60,120,240,360,480,720,960,1200,1440,1680,1920, and 3840 and use 1-hot encoding for each rounded durations.

Algorithm 3 Automatic pseudo melody generation

for each mora in the input-lyrics **do**
  $b \leftarrow$ get boundary type next to the mora
  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_{note}) to the mora
  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

127. For example, the target and its previous notes are 68 and 65, respectively, and the gap is $+3$.

5.3.3 Training Strategies

We propose two training strategies (i.e., *pretraining* and *learning with a pseudo-melody*) to obtain a robust lyrics language model with a limited amount of melody-lyric alignment data.

5.3.3.1 Pretraining

The size of our melody-lyric alignment data is limited. However, we can obtain a large amount of raw lyric texts. We therefore pretrain the model with the raw lyric texts, and then fine-tune it with the melody-lyric 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.

5.3.3.2 Learning with Pseudo-Melody

We propose a method to increase the melody-lyric alignment data by attaching *pseudo melodies* to the obtained raw lyric texts. We generate this pseudo melody by using simple probability distributions. We refer to the model that uses these data as the *Pseudo-melody* model.

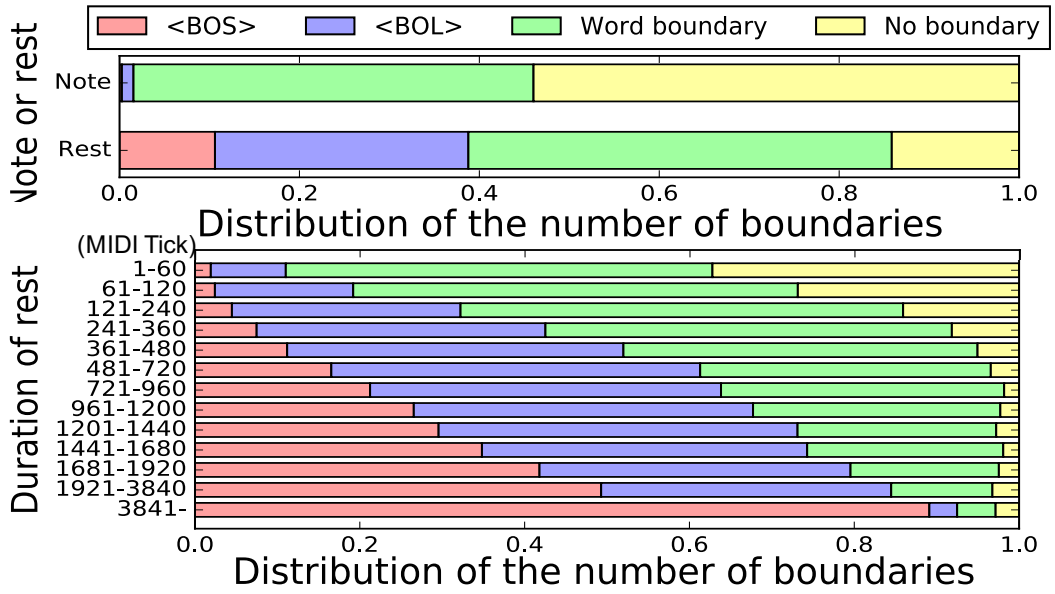


Figure 5.7: Distribution of the number of boundaries in the pseudo data.

Algorithm 3 shows the details of pseudo-melody generation. For each mora in the lyrics, we first assign a note to the mora. Then, we determine whether to generate a rest next to it. Since we already knew the correlations between rests and lyric boundaries, the probability for a rest and its duration is conditioned by a boundary type next to the target mora. The pitch of each note is generated based on the trigram probability. All probabilities are calculated using the training split of the melody-lyric alignment data.

Figure 5.7 shows the distributions of the boundaries in generated pseudo melodies. The distributions closely resembles those of our melody-lyric alignments in Figure 5.5.

5.4 Quantitative Evaluation

We performed experiments to evaluate the quality of the generated lyrics based on evaluation metrics for word fluency and consistency in the lyric segment boundaries.

5.4.1 Experimental Setup

5.4.1.1 Hyperparameters

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 lyric generation, we used a beam search with a width of 10. An example of the generated lyrics is shown in Figure 5.10.

5.4.2 Evaluation Metrics

5.4.2.1 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 and segment boundaries, to investigate the predictability of words.

5.4.2.2 Accuracy of Boundary Replication

Under the assumption that the line and segment 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 performance was measured in terms of the precision, recall, and F_1-measures of the boundary positions.

We also asked a person to manually place line and segment boundaries at plausible positions for randomly selected 10 input melodies that the evaluator have never heard. This individual is not a professional musician but an experienced performer educated on musicology. The bottom part of Table **??** represents the human performance. The last line of Table 5.2 shows the result.

Table 5.1: Performance of *Fine-tuned* models with different fixed parameters (F_1-UB denotes the score for unlabeled matching of line/segment boundaries.).

Model	Input-Word Emb	LSTM	Output Layers	PPL	PPL-W	F_1-⟨BOS⟩	F_1-⟨BOL⟩	F_1-UB
Setting 1	Fixed	Fine-tuned	Fixed	**138.8**	**247.3**	0.177	0.211	0.327
Setting 2	Fixed	Fine-tuned	Fine-tuned	152.2	275.5	0.252	**0.325**	**0.512**
Setting 3	Fine-tuned	Fine-tuned	Fine-tuned	172.1	315.4	**0.266**	0.299	0.470

Table 5.2: Effect of pretraining and learning with pseudo-melody (UB evaluates unlabeled matching of line/segment boundaries.).

	PPL	PPL-W	⟨BOS⟩			⟨BOL⟩			UB		
Model			Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
Lyrics-only	138.0	225.0	0.127	0.111	0.119	0.055	0.063	0.059	0.103	0.110	0.106
Full-data	135.9	222.1	0.082	0.139	0.103	0.055	0.040	0.047	0.099	0.102	0.101
Over-sample	133.9	219.4	0.097	0.106	0.101	0.047	0.053	0.050	0.085	0.095	0.090
Alignment-only	173.3	314.8	0.281	0.343	0.309	0.246	0.330	0.282	0.417	0.545	0.473
Fine-tuned	152.2	275.5	0.225	0.287	0.252	**0.290**	**0.370**	**0.325**	**0.456**	**0.582**	**0.512**
Pseudo-melody	**115.7**	**197.5**	0.348	0.275	0.307	0.263	0.272	0.267	0.436	0.419	0.428
Heuristic	175.8	284.7	**0.379**	**0.367**	**0.373**	0.244	0.234	0.239	0.409	0.384	0.402
Human	-	-	0.814	0.640	0.717	0.676	0.666	0.671	0.780	0.724	0.751

5.4.3 Comparison of Pretraining Settings

In the pretraining strategy, we have multiple options to select parameters for fine-tuning. To explore optimal settings, we evaluated the performance of the *Fine-tuned* models, where some of the pretrained parameters are fixed or fine-tuned. Table 5.1 summarizes the result.

Compared with *Setting 2*, the performance of *Setting 3* in PPL and PPL-W is reduced significantly when the word embedding layer is fine-tuned. This result indicates that fine-tuning word vectors with a small quantity of data corrupts the language model. On the other hand, fixing mora-count and word output layers somewhat limits the benefit from the melody data. Here, the performance of *Setting 1* relative to boundary replication is less than that of *Settings 2* and *3*. For these reasons, we selected *Setting 2* as the optimal setting for the *Fine-tuned* model.

5.4.4 Effect of Melody-conditioned RNNLM

To investigate the effect of our language models, we compared the following seven 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-lyric alignments of those), (3) a *Over-sample* model trained with 54,081 song lyrics and 54,081 melody-lyric alignment (By copying the training instance, we increased the number of melody-lyric alignments to 54,081 and balanced the amount of lyrics data and melody-lyric alignments), and (4) an *Alignment-only* model trained with only 900 melody-lyric alignment data. The remaining two are Melody-conditioned RNNLMs with the proposed learning strategies: (5) *Fine-tuned* and (6) *Pseudo-melody* models. The remaining one is (6) *Heuristic* model that: (i) assigns a line/segment boundary to a rest based on its duration with the same probability as reported in Figure 5.5, and (ii) fills the space between any two boundaries with lyrics of the appropriate mora counts. This *Heuristic* model computes the following word probability:

$$P(w_t|w_0,...,w_{t-1},\mathbf{m}) = \begin{cases} Q(\langle \mathrm{BOS} \rangle | m_{i(t+1)}) & (\text{if } w_t = \langle \mathrm{BOS} \rangle) \\ Q(\langle \mathrm{BOL} \rangle | m_{i(t+1)}) & (\text{if } w_t = \langle \mathrm{BOL} \rangle) \\ (1 - Q(\langle \mathrm{BOS} \rangle | m_{i(t+1)}) - Q(\langle \mathrm{BOL} \rangle | m_{i(t+1)})) \times & \\ \frac{P_{\mathrm{LSTM}}(w_t|w_0,...,w_{t-1})}{1-P_{\mathrm{LSTM}}(\langle \mathrm{BOL} \rangle|w_0,...,w_{t-1})-P_{\mathrm{LSTM}}(\langle \mathrm{BOS} \rangle|w_0,...,w_{t-1})} & (\text{otherwise}) \end{cases} \tag{5.8}$$

where Q is the same probability as reported in Figure 5.5. P_{LSTM} is the word probability calculated by a standard LSTM language model.

Table 5.2 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/segment boundaries (i.e., UB). This result indicates that our Melody-conditioned RNNLMs and the heuristic approach successfully capture the consistency between melody and lyric boundaries. 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 text data and this harms the learning process of the dependency between melody and lyrics. For the segment boundary, the *Heuristic* model achieved the best performances. For the line boundary, on the other hand, the *Fine-tuned* model achieved the best performances.

Regarding PPL and PPL-W, the *Lyrics-only*, *Full-data*, and *Pseudo-melody* mod-

Table 5.3: Effect of mora-count output layer ((w/o $\mathbf{y}_s$) denotes exclusion of the mora-count output layer.).

Model	PPL	PPL-W	F_1-UB
Fine-tuned	**152.2**	**275.5**	**0.512**
Fine-tuned (w/o $\mathbf{y}_s$)	155.1	278.1	0.323
Pseudo-melody	**115.7**	**197.5**	**0.428**
Pseudo-melody (w/o $\mathbf{y}_s$)	118.0	201.5	0.397

els 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, *Pseudo-melody* model achieved even better performance than the *Full-data* model and 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, *Heuristic* model has the worst performance, despite training a large amount of raw lyrics text. The reason why the *Heuristic* model generates non-fluent lyrics is analyzed in Section 5.4.6.

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-lyric 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.

5.4.5 Effect of Predicting Mora-Counts

To investigate the effect of predicting mora-counts, we compared the performance of the proposed models to models that exclude the mora-count output layer $\mathbf{y}_s$. Table 5.3 summarizes the results. For the pretraining strategy, the use of $\mathbf{y}_s$ significantly alleviates data sparsity when learning the correlation between mora 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 relatively competitive in both measures. This means that the *Pseudo-melody* model obtained a sufficient amount of word-

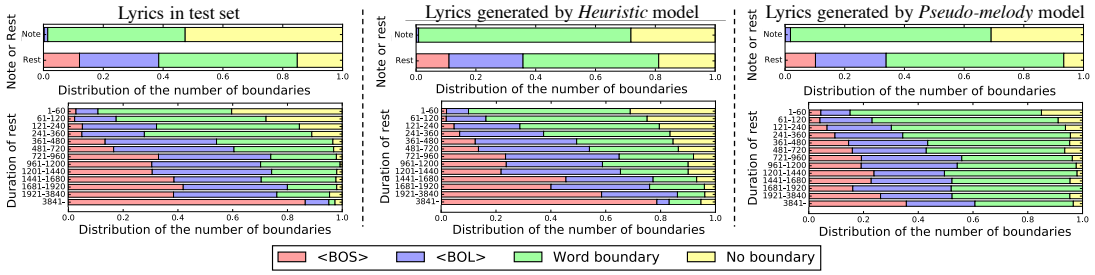


Figure 5.8: Distribution of the number of boundaries in the test data and lyrics generated by the *Pseudo-melody* model.

melody input pairs to learn the correlation.

5.4.6 Analysis of Input Melody and Generated Lyrics

To examine whether the models can capture correlations between rests and lyrics boundaries, we calculate the proportion of the word, line, and segment boundaries in the original lyrics and in the lyrics generated by the *Heuristic* and *Pseudo-melody* model for the test set (Figure 5.8). The proportion of ⟨BOL⟩ and ⟨BOS⟩ generated by the *Heuristic* model are almost equivalent to those of the original lyrics. On the other hand, for the *Pseudo-melody* model, when rests are longer, the proportion of line/segment boundary types are smaller than that for the original lyrics, and there is still room for improvement.

Although the *Heuristic* model reproduces the proportion of the original lyrics boundaries, the model has the low performance for fluent lyrics generation as shown in Section 5.4.4. To investigate this phenomenon, we observe the lyrics generated by the *Heuristic* and find that the model tends to generates line/segment boundaries after the melody rest, even if two rests are placed very close. Figure 5.9 shows that the distributions of the mora count of the lines and segments and the Jensen-Shannon divergence Lin [1991] of the distributions. This figure indicates the distributions of lyrics of the test set are more similar to the distributions of lyrics generated by the proposed *Pseudo-melody* model rather than the *Heuristic* model. This result supports that the heuristic approach, which simply generates line/segment boundaries based on the distribution in Figure 5.5, cannot generate fluent lyrics with well-formed line/segment lengths.

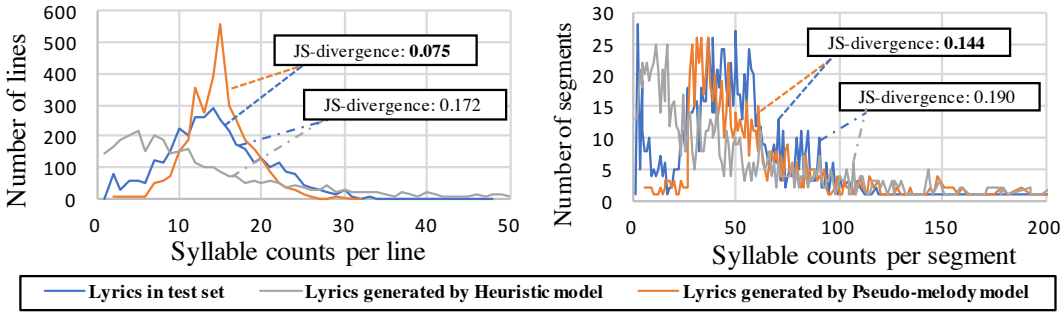


Figure 5.9: Distribution of the mora count of the generated lines/segments.

5.5 Qualitative Evaluation

In order to evaluate how humans feel the lyrics generated by the proposed model, inspired by [Oliveira, 2015], we asked crowdsourcing workers to listen to melody and lyrics, and rate for the following five questions using a five-point Likert scale:

1. **Listenability (L):** when listening to melody and lyrics, are the positions of words, lines, and segments natural? (1=Poorly to 5=Perfectly)
2. **Grammaticality (G):** is the lyric grammatically correct? (1=*Poorly* to 5=*Perfectly*)
3. **Sentence-level meaning (SM):** is each line of lyrics meaningful? (1=*No sense* to 5=*Clear*)
4. **Document-level meaning (DM):** is the entire lyrics meaningful? (1=*No sense* to 5=*Clear*)
5. **Overall quality (OQ):** what is the overall quality of lyrics when listening to music? (1=*Terrible* to 5=*Great*)

5.5.1 Experimental Setup

For the evaluation sets, we randomly selected four melody from the RWC music database [Goto et al., 2002]. For each melody, we prepared five patterns of lyrics to investigate the effect of our language models: lyrics generated by *Heuristic*, *Lyrics-only*, *Fine-tuned*, and *Pseudo-melody* models. The remaining one is lyrics created by amateur human writers: to obtain an upper bound for this evaluation, we asked four Japanese to write lyrics on the evaluation melody. One user was a junior high school teacher of music who had experience in music composition and lyric writing. Three

Table 5.4: Results of the qualitative evaluation.

Metric	*Heuristic*		*Lyrics-only*		*Fine-tuned*		*Pseudo-melody*		*Human* (Upper-bound)	
	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
SM	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

users were graduate students with different levels of musical expertise. Two of them had experience with novel composition, and two had experience with music composition, but none of them had experience with lyric writing.[1] We asked 50 workers to evaluate lyrics for each evaluation song, and obtained 1,000 samples in total. Note that workers did not know whether lyrics was created by human, or generated by computer.

5.5.2 Results

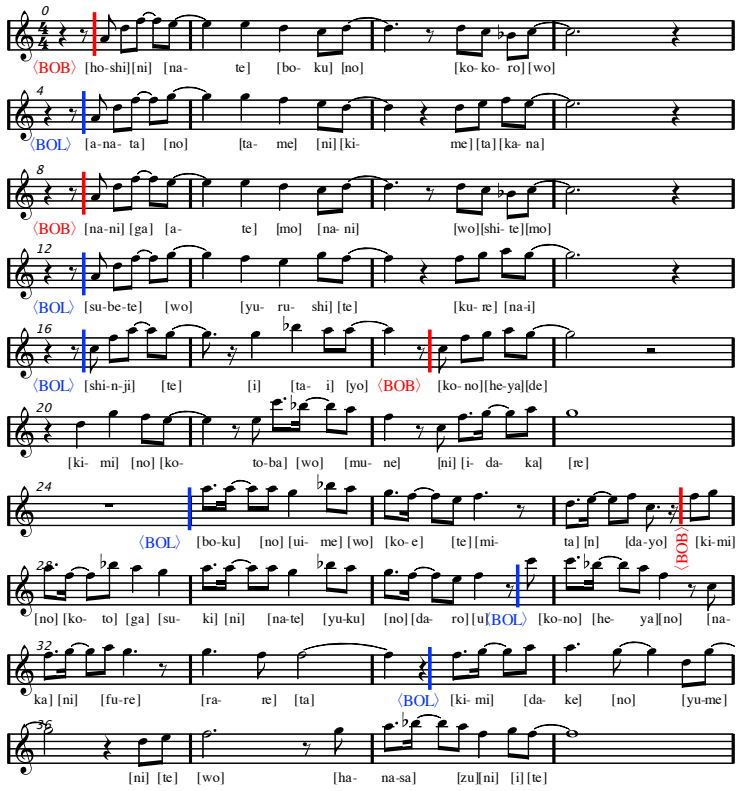
Table 5.4 shows the average scores, standard deviations, and medians each metric assigned to each model for the lyrics used in qualitative evaluation. Regarding the "Listenability" evaluation, workers rated highly the lyrics generated by the *Fine-tuned* model and the *Pseudo-melody* model that are trained on the melody and lyrics. On the other hand, regarding the "Grammar" and "Meaning" evaluation, workers rated highly the lyrics generated by the *Lyrics-only* model and the *Pseudo-melody* model that are well-trained on a large amount of text data. These results are entirely consistent with those of the quantitative evaluations. Regarding "Overall quality" evaluation, Pseudo-melody model outperformed the other models. These results indicates our pseudo data learning strategy contributes to high-quality lyrics generation. However, the quality of lyrics generated automatically is still worse than the quality of humans production, and it remains a challenge for future research to develop computational models that generate more high-quality lyrics.

[1]We publish lyrics and audio files used in qualitative evaluation on the Web `http://www.cl.ecei.tohoku.ac.jp/lyrics`.

5.6 Conclusion

This chapter has presented a novel data-driven approach for building a melody-conditioned lyrics language model. We created a 1,000-song melody-lyric 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-lyric correlations with this size of data. We have also proposed a RNN-based, melody-conditioned language model that generates fluent lyrics whose word/line/segment boundaries fit a given input melody. Our experimental results have shown that: (1) our Melody-conditioned RNNLMs capture the consistency between melody and lyric boundaries while maintaining word fluency; (2) combining a limited-scale collection of melody-lyric alignment data with a far larger collection of lyrics-alone data for training the model boosts the model's competence; and (3) we have also produced positive empirical evidence for the effect of applying a multi-task learning schema where the model is trained for mora count prediction as well as for word prediction. (4) the lyrics generated by the model that learned the pseudo melody-lyrics alignments were highly evaluated in the human assessment.

We need to 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-lyric alignment data enables us to explore such a broad range of aspects of melody-lyric correlations.



Generated Lyrics	Translated Lyrics
ho-shi ni na-te bo-ku no ko-ko-ro wo 星になって 僕の心を a-na-ta no ta-me ni ki-me ta ka-na あなたの為に 決めたかな	I became a star. My heart... I guess I decided it was for you
na-ni ga a-te mo na-ni wo shi-te mo 何があっても 何をしても su-be-te wo yu-ru-shi te ku-re na-i 全てを許してくれない shi-n-ji te i ta-i yo 信じていたいよ	Whatever happens. Whatever I do... You will not forgive everything I want to believe
ko-no he-ya de ki-mi no ko-to-ba wo mu-ne ni i-da-ka re この部屋で 君の言葉を胸に抱かれ bo-ku no yu-me wo ko-e te mi-ta n da-yo 僕の夢を越えてみたんだよ	In this room...I hold your words in my heart. I tried to surpass my own dreams
ki-mi no ko-to ga su-ki ni na-te yu-ku no da-ro u 君の事が好きになってゆくのだろう ko-no he-ya no na-ka ni fu-re ra-re ta この部屋の中に触れられた ki-mi da-ke no yu-me ni te-wo ha-na-sa zu ni i te 君だけの夢に 手を離さずにいて	I will continue to love you I was touched inside of this room Never let go of your dreams

Figure 5.10: An example Japanese lyric generated by the *Pseudo-melody* model. The Japanese lyric is translated into English (the right side). The red and blue lines denote segment and line boundaries, respectively. The song is from the RWC Music Database (RWC-MDB-P-2001 No.20). The generated words and line/segment boundaries are consistent with the melody segments as one can observe that every line/segment boundary is placed immediately after a melody rest. Each line of the lyric is sufficiently fluent as a piece of Japanese lyric. However, there is still room for improvement on the coherence of the entire lyric.

Chapter 6

Interactive Support System for Writing Lyrics

In the previous chapters, we modeled the discourse structure from the viewpoints of repeated patterns, storylines and melodies. This chapter presents our novel Japanese and English lyric-writing support system, "LyriSys" (Figure 6.1) using the discourse structure model. LyriSys can assist a writer in incrementally taking the above factors into account through an interactive interface by generating candidate pieces of lyrics that satisfy the specifications provided by the writer. The capability of automatically generating lyrics and allowing the user to create lyrics incrementally in a trial-and-error manner can be useful for both novices and experts.

6.1 Overview of Writing Support

How a computer system can support writing is an intriguing question. Existing support systems can be classified into two types, systems that can generate entire lyrics automatically [Abe and Ito, 2012; Oliveira, 2015; Settles, 2010] and tools designed to assist the user in searching for words that satisfy a query and usage examples from stored lyrics.[1] However, we believe that neither type of system is appropriate for lyric-writing support. The former type allows users highly limited interaction, whereas the latter requires users to do everything but word search.

[1] MasterWriter. `http://masterwriter.com/`

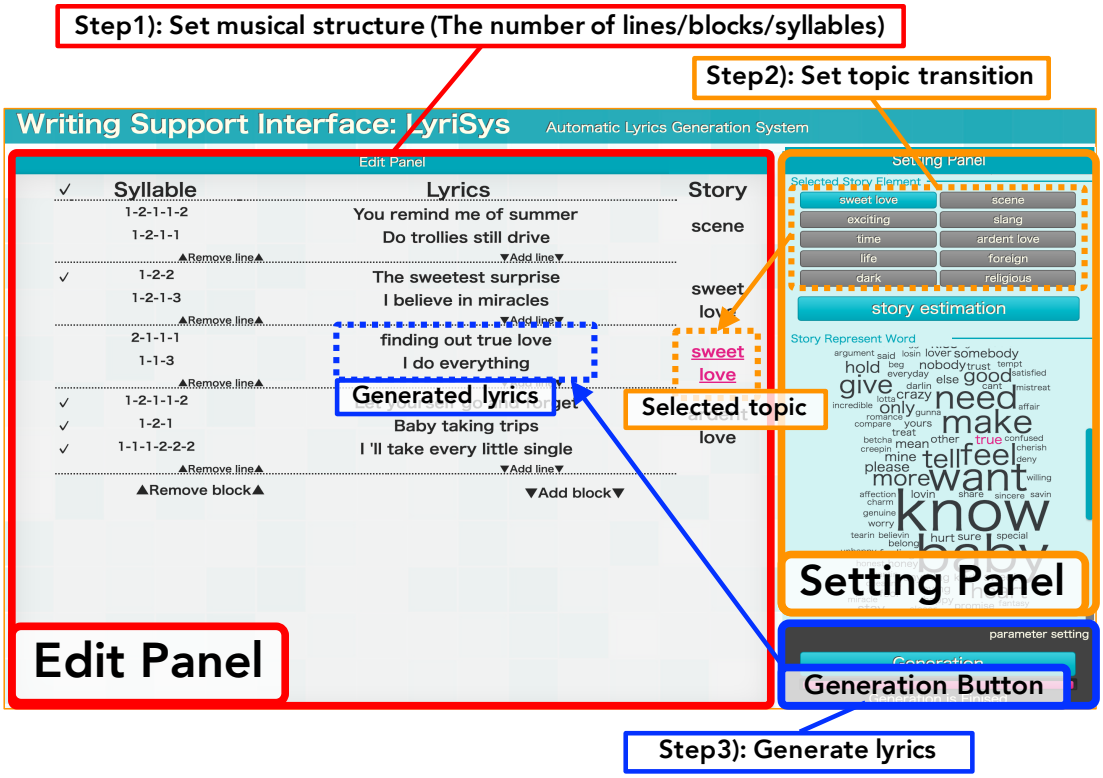


Figure 6.1: An example LyriSys screen.

GarageBand[1] is a system commonly used by beginners as a support tool for song composition. In this interface, the user selects abstract conditions such as ⟨cheerful⟩ and ⟨dark⟩, and then the interface searches for melody patterns that satisfy the specified conditions, thereby enabling novice users to compose a song incrementally by only selecting and setting out the melody pattern with a time line. However, there is no lyric-writing interface for the user to input abstract conditions. In LyriSys, it is relatively easy to create lyrics that represent a storyline by selecting a topic, such as ⟨scene⟩ or ⟨sweet love⟩.

[1]GarageBand. `http://www.apple.com/mac/garageband/`

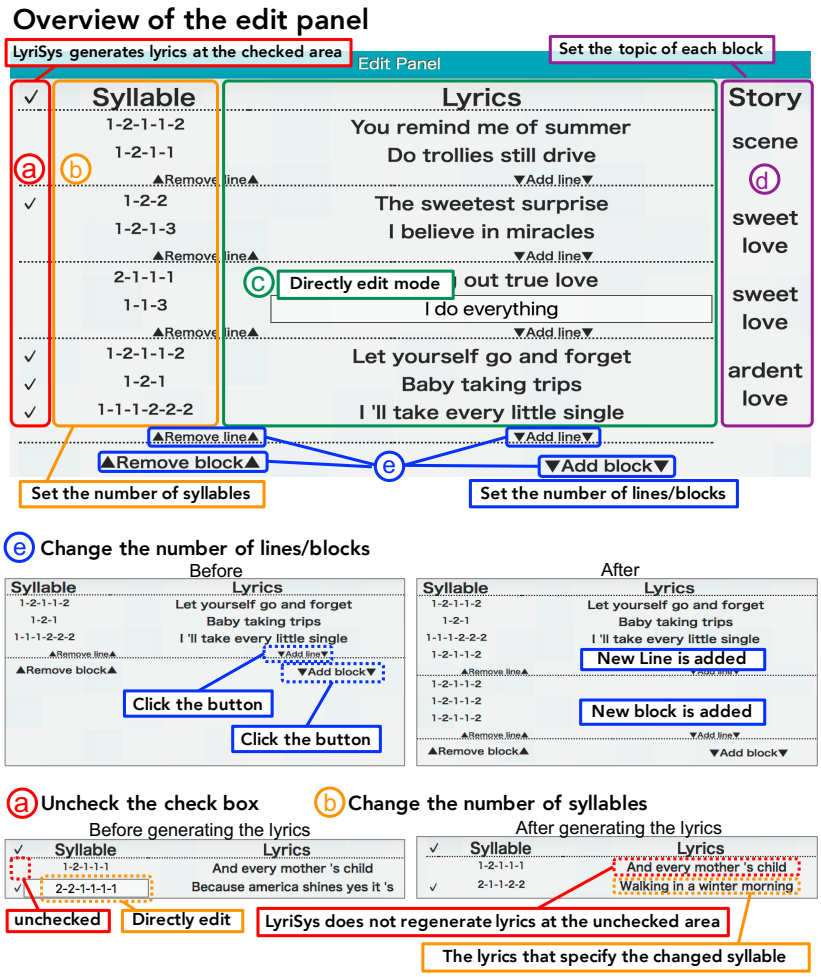


Figure 6.2: The edit panel in LyriSys; the user can set the musical structure, the number of lines/segments/syllables (ⓑ and ⓔ), and the user is allowed to directly edit the current draft using the keyboard (ⓒ).

6.2 LyriSys: An Interactive Writing Interface based on Discourse Structure

This section provides an overview of how a user of LyriSys writes lyrics by interacting with the system. Figure 6.1 shows a screen shot of LyriSys's user interface which consists of the *Edit Panel* and the *Setting Panel*. The Edit Panel displays the current specifications of the musical structure and the current candidate lyrics the user is editing. The Setting Panel is used to choose topics. The basic process is as follows:

Step 1) Input the parameters for specifying the musical structure (the number of segments, the number of lines in each segment, and the number of syllables in each line) manually on the edit panel.

Step 2) For each segment, choose a topic from the predefined set of topics manually. LyriSys can also estimate the topics of a given lyrics automatically. This function enables users to learn the concept of storyline and how it works from examples for, say, their favorite existing lyrics. With this function, the user can use the system also to fill only a small number of lines of a given mostly completed lyric.

Step 3) Click the Generation button, LyriSys then generates the lyrics that correspond to the input syllables and the topic. For example, LyriSys automatically generates "I believe in love" when the user inputs the syllable set "1-2-1-1" and the topic ⟨sweet love⟩. This function assists the user in searching the huge space of word sequences. Moreover, users can revise some lines, if they desire, by selecting a line to revise and then choosing an alternative candidate from the candidates displayed on the Setting Panel.

Along with this process, LyriSys generates candidate lyrics that satisfy the topic and the number of syllables. A crucial property of the design of LyriSys is that it allows the user to specify the constraints incrementally and explore the candidate phrases interactively in a trial-and-error manner.

6.2.1 Step 1): Set the Musical Structure

By clicking on the edit panel (Figure 6.2ⓔ), the user sets the musical structure, the number of lines and the number of segments. The user can also always change the number of syllables for each line (Figure 6.2ⓑ). Changing these parameters is allowed at any time; therefore the user can flexibly revise the musical structure. For example, the user might want to change the musical structure of the second verse slightly, while maintaining the musical structure of the first verse.

Moreover, the user can also disable the regeneration of a particular line by unchecking the check box (Figure 6.2ⓐ). This function allows users to specify the lines they are already satisfied with and seek alternative candidates only for the remaining lines. By gradually disabling the regeneration, the user can complete the writing process

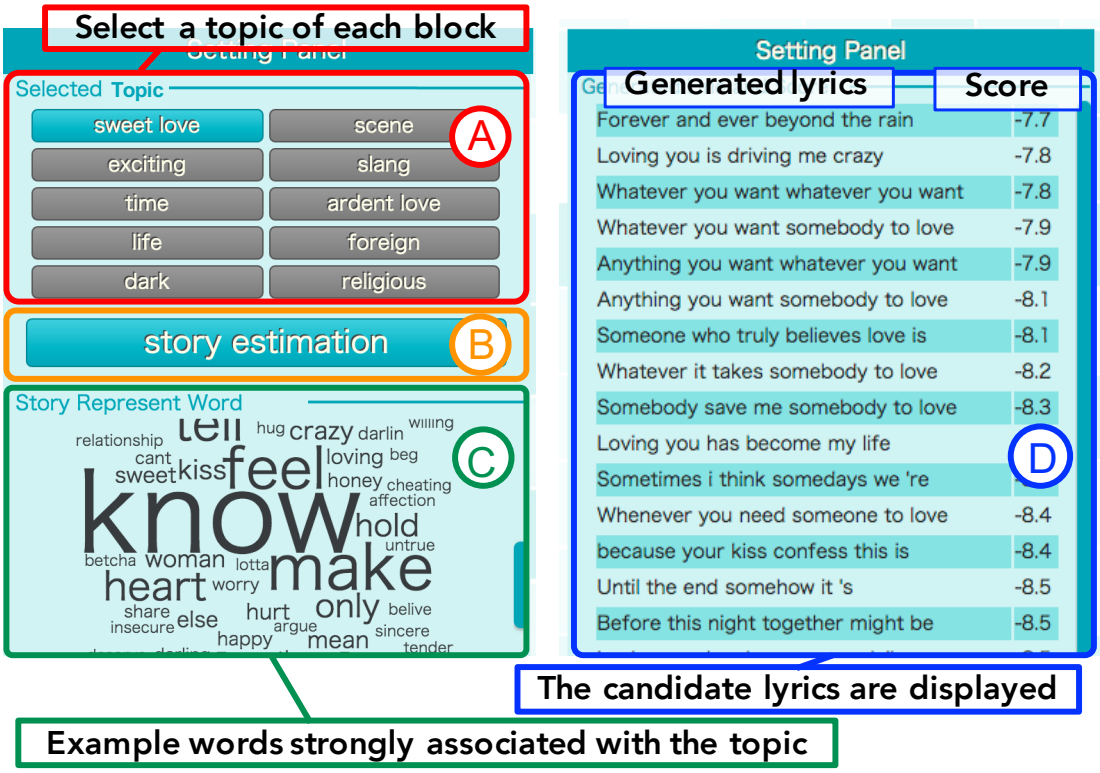


Figure 6.3: The setting panel in LyriSys; the user can set the topic of each segment that displayed in the edit panel (Ⓐ). LyriSys searches for candidate lyrics that satisfy the input parameters, and the user can select a favorite candidate lyrics (Ⓓ).

incrementally.

6.2.2 Step 2): Set/estimate the Story

The process of specifying the topics is as follows: (1) the topic setting panel is displayed (Figure 6.3) by clicking the topic (e.g., ⟨scene⟩) on the edit panel (Figure 6.2ⓓ), and (2) the user selects one topic from the predefined set of topics (Figure 6.3Ⓐ). The setting panel displays example words strongly associated with the topic in question, where the size of each word depicts how likely it is to be chosen for the specified topic (Figure 6.3Ⓒ). This way of displaying the word set is expected to help the user select topics. LyriSys generates a line of lyrics containing as many of these words as possible. For example, when the topic ⟨sweet love⟩ is chosen, words such as "want" are likely to be chosen as in "I want you". It is possible to change the topics at any time,

and the user can revise the generated lyrics. How our probabilistic language model deals with topics is described below in the implementation section.

Some writers might have difficulty in selecting topics. We thus additionally implemented the function of automatically estimating the topics of a given lyric (Figure 6.3Ⓑ). This function is crucially important for users who are not familiar with the notion of topics. First, by applying this function to existing popular lyrics (say, their favorite song lyrics) and seeing the results (i.e., the estimated topics), users can learn what sorts of phrases tend to be generated for each different topic and what transitions over topics can be seen in popular lyrics. Second, users can also begin the writing process by partially rewriting their favorite lyrics, while keeping the original overall structure and topic transitions. Third, modeling topic transitions enables the system to propose a smooth transition of topics for given partly completed draft lyrics.

6.2.3 Step 3): Generate/edit the Candidate Lines of Lyrics

LyriSys searches for candidate lyrics that satisfy the input parameters, when triggered by the generation button, and displays the most probable lyrics in the edit panel (Figure 6.2ⓒ). The user can replace the generated lyrics with other candidates line by line. The candidate lyrics are displayed in the setting panel (Figure 6.3Ⓓ) when selecting a line of the lyrics in the edit panel. The user can select a favorite candidate and click it, resulting in the candidate being displayed in the edit panel. By setting the parameters and selecting candidates repeatedly, the user can gradually compose an entire lyric in a trial-and-error manner. The user is also allowed to edit the current draft directly using the keyboard (Figure 6.2ⓒ).

6.3 Implementation

In this study, to generate lyrics that satisfy the topic and the number of syllables, we calculated tri-gram probabilities and added the number of syllables s and the topic z in the conditions of tri-gram:

$$P(w_i|w_{i-2}, w_{i-1}, s_i, z) = \begin{cases} 0 & (s_i \neq |w_i|) \\ \frac{count(z,w_{i-2},w_{i-1},w_i)}{count(z,w_{i-2},w_{i-1})} & (s_i = |w_i|) \end{cases} \tag{6.1}$$

where $|w_i|$ denotes the number of syllable in a word w_i. This probability is calculated from $count(z, w_{i-2}, w_{i-1}, w_i)$, which returns the number of occurrences of the word string w_{i-2}, w_{i-1}, w_i and the topic z. This value is calculated by counting the number of topics assigned to each segment in Content Model (CM) as mentioned in Chapter 4.2.2.

The enhanced HMM captures the topic transition, which appears in the segment structure. For example $\langle$scene$\rangle \rightarrow \langle$dark$\rangle \rightarrow \langle$sweet love$\rangle$ represents the transition of the topic in three segments. In particular, each z_t is generated from the previous topic z_{t-1} via the transition probability $P(z_t|z_{t-1})$. The word w in each segment is generated from z_t via generative probability $P(w|z_t)$. In addition, it is possible to estimate the topic when uncompleted or unknown lyrics are inputted by using the Viterbi algorithm. Note that the topic z is not labeled as $\langle$sweet love$\rangle$ or $\langle$scene$\rangle$, because z is a latent variable. Therefore, we manually assign labels to each topic by observing the word list whose generative probability $P(w|z)$ is large. Note that the number of topics is changeable before learning; however we set it to ten in this study.

We train the tri-gram and enhanced HMM using unsupervised learning from two datasets that contain Japanese and English lyrics.[1] We used 19,290 Japanese lyrics and 96,475 English lyrics and applied the MeCab part-of-speech parser for Japanese words [Kudo et al., 2004] and Stanford CoreNLP for English words [Manning et al., 2014]. To count the number of syllables, we used a hyphenation algorithm [Liang, 1983]. In lyrics generation, LyriSys searches the word strings so that the lyrics probability $\prod_{i=1}^{n} P(w_i|w_{i-2}, w_{i-1}, s_i, z)$ is large according to the beam search algorithm, where n denotes the number of words in a line.

6.4 User Feedback

To investigate the capabilities, limitations, and potential of our interaction design, we asked five Japanese users to use LyriSys and collected preliminary user feedback. One user was a junior high school teacher of music who had experience in music composition and lyric writing. Four users were graduate students with different levels of musical expertise. Two of them had experience with novel composition, and two had experience with music composition, but none of them had experience with lyric writ-

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

Table 6.1: Results of user feedback: User's positive and negative comments.

Method	Positive Comments	Negative Comments
Baseline Method	*In comparison to other tasks, it was comfortable in not specifying the number of syllable.*	1) *It was difficult to come up with the words that satisfy the melody of song, because of lacking in vocabulary.* 2) *It was difficult to conceive the storyline.*
Previous Method 1	1) *It was easy to write the lyrics because I didn't need to determine which words to use.* 2) *The system sometimes generated cool lyrics pieces without editing manually.*	*I sometimes felt boring because users couldn't edit the generated lyrics.*
Previous Method 2	1) *It was easier to write than the baseline method because the system generated prototype lyrics.* 2) *It was useful to select the candidate of lyrics when the generated result was partially good.*	*It was difficult to write the lyrics that represent the storyline, because only a limited variety of words are generated.*
Proposed Method	1) *It was easy to associate the words related to a topic by viewing the generated candidates of lyrics and the word cloud in the setting panel.* 2) *In comparison to the previous method 2, selecting topics made it easy to write the lyrics that specifies my intention.* 3) *The generated lyrics are more expressive than the result of other interface because of the consideration of topic.*	1) *The list of the 10 topics was too restricted and coarse-grained.* 2) *Although the system generates an abstract storyline, I thought that it would be interesting if the system could generate a concrete storyline.*
Overall Comments	*The support interface was helpful to complete the lyrics particularly when I couldn't come up with any nice words at all.*	1) *The speed of automatic generation was slow.* 2) *I sometimes felt boring because the generated lyrics were same when input parameters were fixed.* 3) *I had a hard time inputting the number of syllable manually.*

ing.

6.4.1 Experimental Setup

We randomly selected five Japanese songs from the RWC music database [Goto et al., 2002] and gave each user one song. Then, we asked the subjects to write lyrics on the melody of the song with the following four tasks.

Baseline method (without interface) In this task, we restricted the use of LyriSys. We gave the subjects a topic transition, e.g., $\langle \text{scene} \rangle \rightarrow \langle \text{sweet love} \rangle \rightarrow \langle \text{positive} \rangle$, and asked the subjects to write the lyrics that satisfy the given topic transition. The purpose of this task is to investigate the difficulty of writing lyrics that sat-

isfy the storyline and the musical melody without the writing support interface.

Previous method 1 (automatic lyrics generation) We asked the subjects to write lyrics with LyriSys, but restricted the use of the selecting/editing of the candidate lines of lyrics. The purpose of this task is to compare the proposed interaction with previous methods that generate an entire piece of lyrics fully automatically [Barbieri et al., 2012; Ramakrishnan A et al., 2009].

Previous method 2 (interaction without topic transition) We implemented another type of LyriSys that has a restricted topic transition function. This LyriSys calculates the simple tri-gram probability $P(w_i|w_{i-2}, w_{i-1}, s_i)$. The purpose of this task is to compare LyriSys with the previous interface, which cannot handle topic transitions [Abe and Ito, 2012].

Proposed method (LyriSys) In this task, we permitted the subjects to use all of the functions of LyriSys.

6.4.2 Results

After the trial usage, we asked the subjects to write comments on each task. Positive and negative comments regarding the capabilities and potential of each task are listed in Table 6.1. These comments suggest that the proposed interface is effective, but that the generation algorithm must be improved to enable the user to write more expressive lyrics.

Figure 6.4 shows an example of lyrics that were created when a user used LyriSys (i.e., the user set the musical structure and the storyline, and selected or edited the recommended lyrics). Moreover, a fully automatically generated lyric is also shown in Figure 6.4. This result shows that the created lyrics correspond to the input parameters (i.e., syllables and topics); for example, we can see the sentimental phrases “ (not show my tears)” and “あなたのそばに (with you)” were created when the topic was ⟨sweet love⟩. Note that these phrases are recommended by LyriSys.

6.5 Conclusion

In this chapter, we proposed a novel lyric-writing interface, LyriSys, that allows users to create and revise their work incrementally in a trial-and-error manner. Through fine-

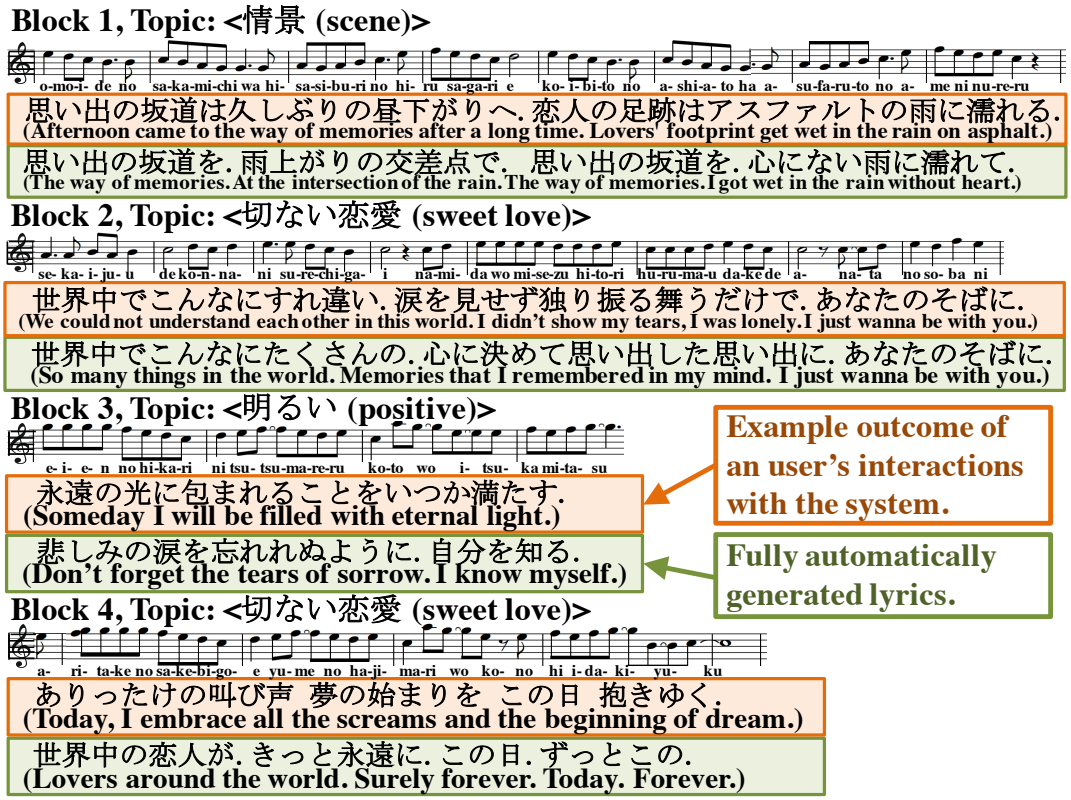


Figure 6.4: Example of Japanese lyrics when the user uses LyriSys. The Japanese lyrics are translated, and the English are given in parentheses. The song is from the RWC Music Database (RWC-MDB-P-2001 No.47).

grained interactions with the system, the user can create the specifications of the music structure (the verse-bridge-chorus structure and the number of lines/syllables) and the transition over topics such as ⟨scene⟩, ⟨dark⟩ and ⟨sweet love⟩.

LyriSys still leaves much room for improvement. It might be too much of a burden for the user to specify the musical structure at the level of syllable counts. This must be relaxed possibly by taking the melody's rhythm directly into account. The present probabilistic language model models semantic topics and topic transitions, but not the verse-bridge-chorus structure, neglecting, for example, the role of choruses. We plan to fix these problems and improve the system by introducing extended functions on the Web.

Chapter 7

Conclusions

This thesis has addressed the issue of modeling discourse structure of lyrics in order to understand and model the discourse-related nature of lyrics. To compute the discourse structure of lyrics, we have addressed four issues:

Does the discourse segments in lyrics strongly correlate with repeating patterns? Phrases of lyrics often appear repeatedly, and this repeated pattern may be correlated with discourse segments. However, no prior study has ever verified this correlation.

What is the most suitable way to model storylines in lyrics? Each discourse segment in lyrics provides part of the entire story and the segments are organized (or sequentially ordered) so as to constitute a coherent structure as a whole. However, no study has ever addressed the issue of modeling storylines in lyrics.

Does the discourse segments in lyrics strongly correlate with melody? Several correlations between melody and lyrics are expected. This direction of research, however, has never been promoted partly because it requires a large training dataset consisting of aligned pairs of lyrics and melody but so far no such data has been available for research.

Are discourse structure models efficient in automatic lyrics generation task? In addition to modeling the discourse structure, we are interested in the effectiveness of discourse model for demonstrating computer systems that automatically generate lyrics or assist human lyricists.

The key contribution of this thesis can be summarized as follows:

1. We proposed a computational model of the discourse segments in lyrics. To test our hypothesis that discourse segmentations in lyrics strongly correlate with repeated patterns, we conduct the first large-scale corpus study on discourse segments in lyrics. This is the first study that takes a data-driven approach to exploring the discourse structure of lyrics in relation to repeated patterns.

2. We proposed computational models to capture the two common discourse-related notions: storylines and themes under the assumption that a storyline is a chain of transitions over topics of segments and a song has at least one entire theme. We tested the hypothesis that considering the notion of theme does contribute to the modeling of storylines of lyrics.

3. We proposed a novel, melody-conditioned lyrics language model and deeply analyzed the correlation between melody and lyrics, This is the first study that has ever provided such strong empirical evidence to the hypotheses about the correlations between lyrics segments and melody rests.

4. We developed a novel interactive support system for writing lyrics. We provides an overview of the design of the system and its user interface and describes how the writing process is guided by our probabilistic discourse structure model.

In Chapter 3, we conducted a large-scale corpus study into the discourse segments of lyrics, in which we examined our primary hypothesis that discourse segmentations strongly correlate with repeated patterns. This is the first study that takes a data-driven approach to explore the discourse structure of lyrics in relation to repeated patterns. We then proposed a task to automatically identify segment boundaries in lyrics and explored machine learning-based models for the task with repeated pattern features and textual features. The results of our empirical experiments show the importance of capturing repeated patterns in predicting the boundaries of discourse segments in lyrics.

In Chapter 4, we presented the first study aiming at capturing the two common discourse-related notions: storylines and themes. We assumed that a storyline is a chain of transitions over topics of segments and a song has at least one entire theme.

We then hypothesized that transitions over topics of lyric segments can be captured by a probabilistic topic model which incorporates a distribution over transitions of latent topics and that such a distribution of topic transitions is affected by the theme of lyrics. Aiming to test those hypotheses, this thesis conducted experiments on the word prediction and segment order prediction tasks exploiting a large-scale corpus of popular music lyrics for both English and Japanese. Our experimental result indicates that typical storylines included in our lyrics datasets were effectively captured as a probabilistic distribution of transitions over latent topics of segments. We can conclude that considering the notion of theme does contribute to the modeling of storylines of lyrics.

In Chapter 5, we presented a novel, data-driven approach for building a melody-conditioned lyrics language model. The model is conditioned with a featurized input melody and trained simultaneously with a mora-count prediction subtask. To build our model and conduct a quantitative investigation into the correlations between melody and lyrics, we actually created a 1,000-song alignment dataset. No prior study has ever conducted such a quantitative analysis of lyrics-melody correlations with this size of data. Our experimental results have shown that combining a limited-scale collection of lyrics-melody alignment data and a far larger collection of lyrics-alone data for training the model boosts the model's competence.

In Chapter 6, we proposed a novel lyric-writing interface, LyriSys, that allows users to create and revise their study incrementally in a trial-and-error manner. Through fine-grained interactions with the system, the user can create the specifications of the music structure (the verse-bridge-chorus structure and the number of lines/moras) and the transition over topics such as ⟨scene⟩, ⟨dark⟩ and ⟨sweet love⟩.

Appendix A

Proof of Theorem in Chapter 4.2

A.1 Equation for Mixture of Unigram and Content Model Inference

$$P(y_m = i|\mathbf{y}_{\neg m}, \mathbf{w}, \epsilon, \zeta) \propto (M_{i,\neg m} + \epsilon) \cdot \frac{\Gamma(N_{i,\neg m} + \zeta V)}{\Gamma(N_{i,\neg m} + N_m + \zeta V)} \cdot \prod_{v:N_{m,v}>0} \frac{\Gamma(N_{i,v,\neg m} + N_{m,v} + \zeta)}{\Gamma(N_{i,v,\neg m} + \zeta)} \tag{A.1}$$

$$P(z_{m,s} = j|\mathbf{z}_{\neg(m,s)}, \mathbf{w}, \alpha, \beta) \propto \frac{S_{z_{m,s-1}\to j,\neg(m,s)} + \alpha}{S_{z_{m,s-1}\to *,\neg(m,s)} + \alpha J} \cdot \frac{S_{j\to z_{m,s+1},\neg(m,s)} + \mathbb{1}(z_{m,s-1} = j = z_{m,s+1}) + \alpha}{S_{j\to *,\neg(m,s)} + \mathbb{1}(z_{m,s-1} = j) + \alpha J} \cdot \frac{\Gamma(N_{j,\neg(m,s)} + \beta V)}{\Gamma(N_{j,\neg(m,s)} + N_{(m,s)} + \beta V)} \cdot \prod_{v:N_{(m,s),v}>0} \frac{\Gamma(N_{j,v,\neg(m,s)} + N_{(m,s),v} + \beta)}{\Gamma(N_{j,v,\neg(m,s)} + \beta)} \tag{A.2}$$

$$P(x_{m,s,n} = k|\mathbf{x}_{\neg(m,s,n)}, \mathbf{w}, \eta, \zeta, \beta) \propto \frac{N_{m,k,\neg(m,s,n)} + \eta}{N_{m,\neg(m,s,n)} + 2\eta} \cdot \Big(\frac{N_{y_m,w_{m,s,n},\neg(m,s,n)} + \zeta}{N_{y_m,\neg(m,s,n)} + \zeta V}\Big)^{1-k} \cdot \Big(\frac{N_{z_{m,s},w_{m,s,n},\neg(m,s,n)} + \beta}{N_{z_{m,s},\neg(m,s,n)} + \beta V}\Big)^{k} \tag{A.3}$$

The update equations in Algorithm 1 can be rewritten as Eq. A.1, A.2 and A.3. Table A.1 shows the notations in Eq. A.1 for collapsed Gibbs sampling of theme y in the MUM-CM inference. Table A.2 shows the notations in Eq. A.2 for collapsed

Table A.1: Notations in Eq. A.1 for MUM-CM

Notation	Definition
$\Gamma(\cdot)$	Gamma function
ϵ, ζ	Hyperparameter
$\mathbf{w}$	Word set in training corpus
$\mathbf{y}_{\neg m}$	Theme set except the m-th lyric
V	Size of the vocabulary
$M_{i,\neg m}$	# of lyrics with theme label i except the m-th lyric
N_m	# of words in the m-th lyric
$N_{i,\neg m}$	# of the word whose theme label is i except the m-th lyric
$N_{m,v}$	# of a word v in the m-th lyric
$N_{i,v,\neg m}$	# of a word v whose theme label is i except the m-th lyric

Gibbs sampling of topic z in the MUM-CM inference. Table A.3 shows the notations in Eq. A.3 for collapsed Gibbs sampling of binary variable x in the MUM-CM inference.

A.2 Equation for Mixture of Content Model Inference

$$P(y_m = i|\mathbf{y}_{\neg m}, \mathbf{z}, \alpha, \epsilon) \propto (M_{i,\neg m} + \epsilon)$$
$$\cdot \prod_{s=1}^{S_m} \left(\frac{\Gamma(S_{i,z_{m,s}\to *,\neg m} + \alpha J)}{\Gamma(S_{i,z_{m,s}\to *,\neg m} + S_{m,z_{m,s}\to *} + \alpha J)} \cdot \prod_{s'=1}^{S_m} \frac{\Gamma(S_{i,z_{m,s}\to z_{m,s'},\neg m} + S_{m,z_{m,s}\to z_{m,s'}} + \alpha)}{\Gamma(S_{i,z_{m,s}\to z_{m,s'},\neg m} + \alpha)} \right) \tag{A.4}$$

The update equation in Algorithm 2 can be rewritten as Eq. A.4. Table A.4 shows the notations in Eq. A.4 for collapsed Gibbs sampling of theme y in the MCM inference.

Table A.2: Notations in Eq. A.2 for MUM-CM

Notation	Definition
$\mathbb{1}(\cdot)$	Indicator function
α, β	Hyperparameter
$\mathbf{w}$	Word set in training corpus
$\mathbf{z}_{\neg(m,s)}$	Topic set except the s-th segment in the m-th lyric
J	# of topics
$S_{z_{m,s-1}\to j,\neg(m,s)}$	# of segments that trans topic $z_{m,s-1}$ to j except the s-th segment in the m-th lyric
$S_{z_{m,s-1}\to *,\neg(m,s)}$	# of segments with topic $z_{m,s-1}$ except the s-th segment in the m-th lyric
$N_{(m,s)}$	# of words in the s-th segment in the m-th lyric
$N_{j,\neg(m,s)}$	# of words whose topic label is j except the s-th segment in the m-th lyric
$N_{(m,s),v}$	# of a word v in the s-th segment in the m-th lyric
$N_{j,v,\neg(m,s)}$	# of a word v whose topic label is j except the s-th segment in the m-th lyric

Table A.3: Notations in Eq. A.3 for MUM-CM

Notation	Definition
V	Size of the vocabulary
η, ζ, β	Hyperparameter
$\mathbf{w}$	Word set in training corpus
$\mathbf{x}_{\neg(m,s,n)}$	Binary variable set except the n-th binary variable of the s-th segment in the m-th lyric
$N_{m,\neg(m,s,n)}$	# of words in the m-th lyric except the n-th word of the s-th segment in the m-th lyric
$N_{m,k,\neg(m,s,n)}$	# of words in the m-th lyric with binary label k except the n-th word of the s-th segment in the m-th lyric
$N_{y_m,\neg(m,s,n)}$	# of a word whose theme label is y_m except the n-th binary variable of the s-th segment in the m-th lyric
$N_{y_m,w_{m,s,n},\neg(m,s,n)}$	# of a word $w_{m,s,n}$ with theme label y_m except the n-th binary variable of the s-th segment in the m-th lyric
$N_{z_{m,s},\neg(m,s,n)}$	# of a word whose topic label is $z_{m,s}$ except the n-th binary variable of the s-th segment in the m-th lyric
$N_{z_{m,s},w_{m,s,n},\neg(m,s,n)}$	# of a word $w_{m,s,n}$ with topic label $z_{m,s}$ except the n-th binary variable of the s-th segment in the m-th lyric

Table A.4: Notations in Eq. A.4 for MCM

Notation	**Definition**
α, ϵ	Hyperparameter
$\mathbf{z}$	Topic set in training corpus
$\mathbf{y}_{\neg m}$	Theme set except the m-th lyric
J	# of topics
$M_{i,\neg m}$	# of lyrics with theme label i except the m-th lyric
$S_{m,z\to *}$	# of segments with topic z in the m-th lyric
$S_{y,z\to *,\neg m}$	# of segments whose topic is z and theme is y except the m-th lyric
$S_{m,z\to z'}$	# of segments whose topic transitions z to z' in the m-th lyric
$S_{y,z\to z',\neg m,}$	# of segments whose theme is y and topic transitions z to z' in the m-th lyric except the m-th lyric

References

Ananth Ramakrishnan A and Sobha Lalitha Devi. An alternate approach towards meaningful lyric generation in Tamil. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, pages 31–39, 2010. 1

Chihiro Abe and Akinori Ito. 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, 2012. 1, 9, 64, 72

Dave Austin, Jim Peterik, and Cathy Lynn Austin. *Songwriting for Dummies*. Wileys, 2010. 1, 11, 12, 24, 40

Adriano Barate, Luca Andrea Ludovico, and Enrica Santucci. A semantics-driven approach to lyrics segmentation. In *Proceedings of the 8th International Workshop on Semantic and Social Media Adaptation and Personalization*, pages 73–79, 2013. 8, 11

Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. Markov constraints for generating lyrics with style. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012)*, pages 115–120, 2012. 1, 8, 72

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT 2004)*, pages 113–120, 2004. 26, 28

Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Journal of Machine learning*, 34(1):177–210, 1999. 10, 18, 19, 20

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine learning (ICML 2006)*, pages 113–120, 2006. 26

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 8, 24

Freddy Y.Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33, 2000. 10

Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of International Conference on Multimedia and Expo 2000*, pages 452–455, 2000. 11

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1183–1191, 2016. 1

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*, 14(5):1783–1794, 2006. 11, 42

Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proceedings of the 3rd of International Society for Music Information Retrieval (ISMIR 2002)*, volume 2, pages 287–288, 2002. 2, 3, 11, 21, 22, 60, 71

Erica Greene, Tugba Bodrumlu, and Kevin Knight. 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, 2010. 1, 2, 7

Barbara J. Grosz and Candace L. Sidner. Attention, intentions, and the structure of discourse. *Computational linguistics*, 12(3):175–204, 1986. 24

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 51

Sergey Ioffe and Christian Szegedy. 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, 2015. 52

Tomoharu Iwata, Shinji Watanabe, Takeshi Yamada, and Naonori Ueda. Topic tracking model for analyzing consumer purchase behavior. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI 2009)*, volume 9, pages 1427–1432, 2009. 25

Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. Linear segmentation and segment significance. In *Proceedings of the 6th International Workshop of Very Large Corpora (WVLC-6)*, pages 197–205, 1998. 11

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014. 55

Florian Kleedorfer, Peter Knees, and Tim Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*, pages 287–292, 2008. 8

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 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, 2004. 34, 45, 70

Mirella Lapata. Automatic evaluation of information ordering: Kendall's tau. *Computational Linguistics*, 32(4):471–484, 2006. 35

Franklin Mark Liang. *Word Hy-phen-a-tion by Com-put-er*. Citeseer, 1983. 70

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991. 59

Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 275–282, 2004. 11

Jose P. G. Mahedero, Álvaro MartÍnez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475–478, 2005. 2

Igor Malioutov and Regina Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 25–32, 2006. 10

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL 2014) System Demonstrations*, pages 55–60, 2014. 34, 70

Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*, pages 337–342, 2008. 1, 2, 7

Brian McFee and Daniel PW Ellis. Learning to segment songs with ordinal linear discriminant analysis. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing 2014*, pages 5197–5201, 2014. 11

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proceedings of Interspeech 2010*, pages 1045–1048, 2010. 5, 43, 51

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119. 2013. 40

Thomas P. Minka. Estimating a dirichlet distribution. Technical report, 2000. URL http://research.microsoft.com/en-us/um/people/minka/papers/dirichle 31, 33

Saul B. Needleman and Christian D. Wunsch. 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, 1970. 45

Eric Nichols, Dan Morris, Sumit Basu, and Christopher Raphael. Relationships between lyrics and melody in popular music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 471–476, 2009. 1, 2, 8, 43

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000. 29

Naoaki Okazaki. Classias: A collection of machine-learning algorithms for classification, 2009. http://www.chokkan.org/software/classias/. 19

Hugo Gonçalo Oliveira. Tra-la-Lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence*, 6(1):87–110, 2015. 9, 60, 64

Hugo R. Goncalo Oliveira, F. Amialcar Cardoso, and Francisco C. Pereira. Tra-la-lyrics: an approach to generate text based on rhythm. In *Proceedings of 4th International Joint Workshop on Computational Creativity*, pages 47–55, 2007. 1, 2, 8, 9

Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 59–68, 2006. 11

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, 2014. 41

Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002. 19

Peter Potash, Alexey Romanov, and Anna Rumshisky. 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, 2015. 1, 2, 8

Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi. Automatic generation of tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 40–46, 2009. 1, 9, 72

Sravana Reddy and Kevin Knight. Unsupervised discovery of rhyme schemes. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 77–82, 2011. 1, 2, 7

Jeffrey C. Reynar. Statistical models for topic segmentation. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 357–364. Association for Computational Linguistics, 1999. 11

Martin Riedl and Chris Biemann. Topictiling: a text segmentation algorithm based on lda. In *Proceedings of Association for Computational Linguistics 2012 Student Research Workshop*, pages 37–42. Association for Computational Linguistics, 2012. 10, 22

Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL HLT 2010)*, pages 172–180, 2010. 26, 35

Shoto Sasaki, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto, and Shigeo Morisihima. 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)*, pages 585–590, 2014. 8

Steven L Scott. Bayesian methods for hidden markov models. *Journal of the American Statistical Association*, pages 337–351, 2002. 33

Burr Settles. Computational creativity tools for songwriters. In *Proceedings of the NAACL-HLT Workshop on Computational Approaches to Linguistic Creativity*, pages 49–57, 2010. 9, 64

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180, 2003. 19

Tatsuji Ueda. *The writing lyrics textbook which is easy to understand (in Japanese).* YAMAHA music media corporation, 2010. 1, 11, 12, 24, 40, 47

Dekai Wu, Karteek Addanki, Markus Saers, and Meriem Beloucif. Learning to freestyle: Hip hop challenge-response induction via transduction rule segmentation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 102–112, 2013. 2

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003. 8

Li Yujian and Liu Bo. A normalized Levenshtein distance metric. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007. 13

Ke Zhai and Jason D. Williams. Discovering latent structure in task-oriented dialogues. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 36–46, 2014. 26, 35

List of Publications

Journal Papers (Refereed)

1. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, and Masataka Goto. Modeling Storylines in Lyrics. IEICE Transaction. on Information and Systems. December 2017.

2. Keita Nabeshima, Kento Watanabe, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. Extracting False Information on Twitter and Analyzing its Diffusion Processes by using Linguistic Patterns for Correction (in Japanese). Journal of Natural Language Processing, Vol.13, No.2, pp.461–484, June 2013.

International Conference/Workshop Papers (Refereed)

1. Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. A Melody-conditioned Lyrics Language Model. In Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018).

2. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. LyriSys: An Interactive Support System for Writing Lyrics Based on Topic Transition. In Proceedings of the 22nd Annual Meeting of the Intelligent User Interfaces Community (ACM IUI 2017), pp.559–563, March 2017.

3. Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan B. L. Smith, and Masataka Goto. Modeling Discourse Segments in Lyrics Using Repeated Patterns. In

Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), pp.1959–1969, December 2016.

4. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. Modeling Structural Topic Transitions for Automatic Lyrics Generation. In Proceedings of the 28th Pacific Asia Conference on Language,Information and Computing (PACLIC 2014), pp.422–431, December 2014.

5. Naoaki Okazaki, Keita Nabeshima, Kento Watanabe, Junta Mizuno and Kentaro Inui. Extracting and Aggregating False Information from Microblogs. In Proceedings of the Workshop on Language Processing and Crisis Information 2013 (LPCI 2013), pp.36–43, October 2013.

Awards

1. The 231th IPSJ SIG (Special Interest Groups) on Natural Language Processing Student Incentive Award (2017)

2. The Association for Natural Language Processing Best Paper Award (2013)

3. The 75th Annual Meeting of the IPSJ Conference Student Incentive Award (2013)

Other Publications (Not refereed)

1. Reina Akama, Sho Yokoi, Kento Watanabe, and Kentaro Inui. Semi-supervised Learning of Style Vectors (in Japanese). In JSAI SIG Technical Reports, Vol. SIG-SLUD-B508-27, pp.96–97, October 2017.

2. Reina Akama, Kento Watanabe, Sho Yokoi, and Kentaro Inui. Unsupervised Learning of Style Space and Style Controllable Dialog System (in Japanese). The 12th NLP Symposium for Young Researchers, September 2017.

3. Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Tomoyasu Nakano, Masataka Goto, and Kentaro Inui. Automatic Lyrics Generation based on Correction between Lyrics and Melodies (in Japanese). In IPSJ SIG Technical Reports, Vol. 2017-NL-231 (15), pp.1-8, May 2017.

4. Shun Kiyono, Ran Tian, Kento Watanabe, Naoaki Okazaki, Kentaro Inui. Analysis of Tense Information for Discourse Relationship Recognition (in Japanese). In Proceedings of the 23rd Annual Meeting of the Association for Natural Language Processing, pp.827–830, March 2017.

5. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, and Masataka Goto. Modeling Theme and Story Line for Lyrics (in Japanese). In Proceedings of the 30th Annual Conference of the Japanese Society for Artificial Intelligence, May 2016.

6. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. LyriSys: Writing Support Interface based on the Grobal Structure of Lyrics (in Japanese). In Proceedings of the 23rd Workshop on Interactive Systems and Software (WISS 2015), December 2015.

7. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, Tomoyasu Nakano, Satoru Fukayama, and Masataka Goto. Automatic Lyrics Generation System Considering Storylines (in Japanese). The 10th NLP Symposium for Young Researchers, September 2015.

8. Shun Kiyono, Kento Watanabe, Naoaki Okazaki, Kentaro, Inui. Automatic Japanese Palindrome Generation Based on Phrase Bi-grams and Transformation Rules (in Japanese). In Proceedings of the 29th Annual Conference of the Japanese Society for Artificial Intelligence, June 2015.

9. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. Modeling Latent Topic Transitions from Large Scale Lyrics Data (in Japanese). In Proceedings of the 77th Annual Meeting of the Information Processing Society of Japan, Vol.2, pp.371–372, March 2015.

10. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. Towards Practical Automatic Lyrics Generation (in Japanese). The 9th NLP Symposium for Young Researchers, September 2014.

11. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. Automatic Lyrics Generation System Considering Global Structure (in Japanese).

In Proceedings of the 20th Annual Meeting of the Association for Natural Language Processing, pp.694–697, March 2014.

12. Kento Watanabe, Naoaki Okazaki, and Kentaro Inui. Exploring Social Problems from Twitter (in Japanese). The 8th NLP Symposium for Young Researchers, September 2013.

13. Kento Watanabe, Keita Nabeshima, Junta Mizuno, and Kentaro Inui. Automatic Classification of False and Correction Information on Twitter (in Japanese). In Proceedings of the 19th Annual Meeting of the Association for Natural Language Processing, pp.178–181, March 2013.

14. Kento Watanabe, Keita Nabeshima, Junta Mizuno, Naoaki Okazaki, and Kentaro Inui. Visualization of Diffusion and Convergence Process of False Information on Twitter (in Japanese). In Proceedings of the 75th Annual Meeting of the Information Processing Society of Japan, Vol.1, pp.657–658, March 2013.