LyriSys: An Interactive Support System for Writing Lyrics Based on Topic Transition

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

ABSTRACT

This paper presents LyriSys, a novel lyric-writing support system. Previous systems for lyric writing can fully automatically only generate a single line of lyrics that satisfies given constraints on accent and syllable patterns or an entire lyric. In contrast to such systems, LyriSys allows users to create and revise their work incrementally in a trial-and-error manner. Through fine-grained interactions with the system, the user can create the specifications of the musical structure and the story of the lyrics in terms of the verse-bridge-chorus structure, the number of lines, words and syllables, and most importantly, the transition over semantic topics such as ⟨scene⟩, ⟨dark⟩ and ⟨sweet love⟩. This paper provides an overview of the design of the system and its user interface and describes how the writing process is guided by a state-of-the-art probabilistic generative topic model that is trained without supervision. The system works for both Japanese and English.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g., HCI): Natural language; H.5.2. Information Interfaces and Presentation (e.g., HCI): Training, help, and documentation.

Author Keywords

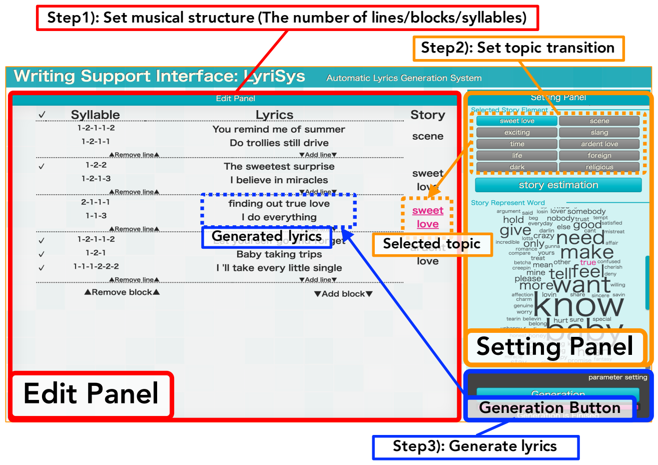
Computational creativity; linguistic creativity; song lyrics; semantic; topic transition.

INTRODUCTION

In popular music, lyrics are an important element in conveying emotions and messages. The recent increasing popularity of consumer generated media makes writing lyrics increasingly popular even for novice users. In writing lyrics, the writer considers various desirable factors, as follows [2, 12]:

Syllable In the case of writing lyrics for a given piece of music, the writer must select words whose syllables correspond to the melody notes.

Rhyming and refrain Lyrics have their own specific properties such as frequent repetitions of identical or similar phrases and extensive use of rhymes and refrains.


Step1): Set musical structure (The number of lines/blocks/syllables)
Step2): Set topic transition
Writing Support Interface: LyriSys
Automatic Lyrics Generation System
Edit Panel
Setting Panel
story estimation
Story Represent Word
Generated lyrics
Selected topic
Edit Panel
Setting Panel
Generation Button
Step3): Generate lyrics

✓	Syllable	Lyrics	Story
	1-2-1-1-2	You remind me of summer	scene
	1-2-1-1	Do trollies still drive	
✓	1-2-2	The sweetest surprise	sweet love
	1-2-1-3	I believe in miracles	
	2-1-1-1	finding out true love	sweet love
	1-1-3	I do everything	
✓	1-2-1-1-2		
✓	1-2-1	Baby taking trips	love
✓	1-1-1-2-2-2	I 'll take every little single	

Figure 1. An example LyriSys screen.

Story Popular music tends to be structured by components called "verses", "bridges" and "choruses". In this paper, we refer to such components as **blocks**. Each piece of lyrics corresponding to a block tends to conveys a particular **topic** such as ⟨scene⟩ or ⟨sweet love⟩, and the manner in which one topic transition to another is important in writing coherent lyrics. In this paper, we refer to the sequence of topics in a song as a **story**.

For writers of lyrics, however, considering these factors simultaneously is not easy. This is where appropriate human-computer interaction system can reduce human loads. This paper presents our novel Japanese and English lyric-writing support system, "LyriSys" (Figure 1). 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.

RELATED WORK

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 [9, 10, 8] and tools designed to assist the user in searching for words that satisfy a query and usage examples from stored

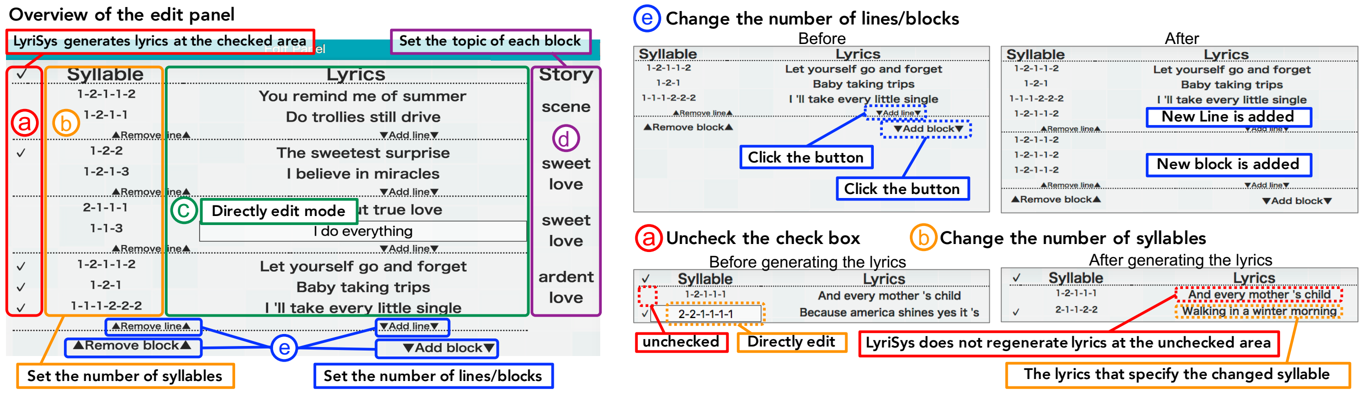

Overview of the edit panel
LyriSys generates lyrics at the checked area
Set the topic of each block
Syllable
Lyrics
Story
1-2-1-1-2
You remind me of summer
1-2-1-1
Do trollies still drive
scene
1-2-2
The sweetest surprise
1-2-1-3
I believe in miracles
sweet love
2-1-1-1
Directly edit mode
1-1-3
I do everything
1-2-1-1-2
Let yourself go and forget
1-2-1
Baby taking trips
1-1-1-2-2-2
I 'll take every little single
ardent love
▲Remove line▲
▼Add line▼
▲Remove block▲
▼Add block▼
Set the number of syllables
Set the number of lines/blocks
Change the number of lines/blocks
Before
After
Click the button
New Line is added
New block is added
Uncheck the check box
Change the number of syllables
Before generating the lyrics
After generating the lyrics
And every mother 's child
Because america shines yes it 's
Walking in a winter morning
unchecked
Directly edit
LyriSys does not regenerate lyrics at the unchecked area
The lyrics that specify the changed syllable

Figure 2. The edit panel in LyriSys; the user can set the musical structure, the number of lines/blocks/syllables (ⓑ and ⓔ), and the user is allowed to directly edit the current draft using the keyboard (ⓒ).

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.

GarageBand[2] 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 story by selecting a topic, such as ⟨scene⟩ or ⟨sweet love⟩.

Automatic lyric generation

Prior studies report systems that can generate only a single line of lyrics independently of the rest [1] but none of the previous systems can consider the global coherence of lyrics such as the topics of blocks and the similarity between and transitions over blocks. Several research groups have proposed lyric generation systems that are designed to generate an entire lyric fully automatically [3, 11]. However, such fully automatic systems are not appropriate for the purpose of writing support; they provide no functionality for the user to control the writing process interactively. In contrast, we implement an interface that allows users to specify what they want and consider the global coherence of lyrics incrementally and interactively.

LYRISYS: AN INTERACTIVE WRITING INTERFACE BASED ON GLOBAL STRUCTURE

This section provides an overview of how a user of LyriSys writes lyrics by interacting with the system. Figure 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:

[1]MasterWriter. http://masterwriter.com/
[2]GarageBand. http://www.apple.com/mac/garageband/

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

Step 2) For each block, 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 story 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.

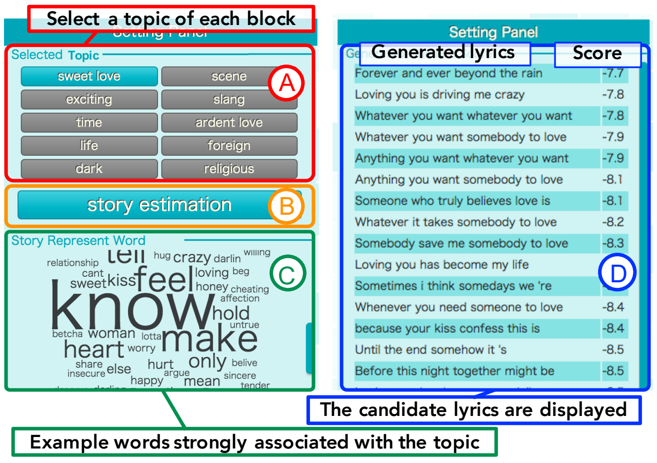
Step 3) Click the Generation button, LyriSys then generates the lyrics that correspond to the input syllables and the topic. For example, LyriSys automatically generates "I believe in love" when the user inputs the syllable set "1-2-1-1" and the topic ⟨sweet love⟩. This function assists the user in searching 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.

Step 1): Set the musical structure

By clicking on the edit panel (Figure 2ⓔ), the user sets the musical structure, the number of lines and the number of blocks. The user can also always change the number of syllables for each line (Figure 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 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 incrementally.



Generated lyrics	Score
Forever and ever beyond the rain	-7.7
Loving you is driving me crazy	-7.8
Whatever you want whatever you want	-7.8
Whatever you want somebody to love	-7.9
Anything you want whatever you want	-7.9
Anything you want somebody to love	-8.1
Someone who truly believes love is	-8.1
Whatever it takes somebody to love	-8.2
Somebody save me somebody to love	-8.3
Loving you has become my life	[illegible]
Sometimes i think somedays we 're	[illegible]
Whenever you need someone to love	-8.4
because your kiss confess this is	-8.4
Until the end somehow it 's	-8.5
Before this night together might be	-8.5

Figure 3. The setting panel in LyriSys; the user can set the topic of each block 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 (Ⓓ).

Step 2): Set/estimate the story

The process of specifying the topics is as follows: (1) the topic setting panel is displayed (Figure 3) by clicking the topic (e.g., ⟨scene⟩) on the edit panel (Figure 2ⓓ), and (2) the user selects one topic from the predefined set of topics (Figure 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 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 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.

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 2ⓒ). The user can replace the generated lyrics with other candidates line by line. The candidate lyrics are displayed in the setting panel (Figure 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 2ⓒ).

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:

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

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 block in an enhanced hidden Markov model (HMM) [13].

The enhanced HMM captures the topic transition, which appears in the block structure. For example ⟨scene⟩ $\rightarrow$ ⟨dark⟩ $\rightarrow$ ⟨sweet love⟩ represents the transition of the topic in three blocks. 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 block 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 ⟨sweet love⟩ or ⟨scene⟩, 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.[3] We used 19,290 Japanese lyrics and 96,475 English lyrics and applied the MeCab part-of-speech parser for Japanese words [5] and Stanford CoreNLP for English words [7]. To count the number of syllables, we used a hyphenation algorithm [6]. 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.

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 writing.

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

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 story.*
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 story, 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 story, I thought that it would be interesting if the system could generate a concrete story.*
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.*

Table 1. Results of user feedback: User's positive and negative comments.

Experimental setup

We randomly selected five Japanese songs from the RWC music database [4] 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., $\langle$scene$\rangle \rightarrow \langle$sweet love$\rangle \rightarrow \langle$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 satisfy the story 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 [3, 11].

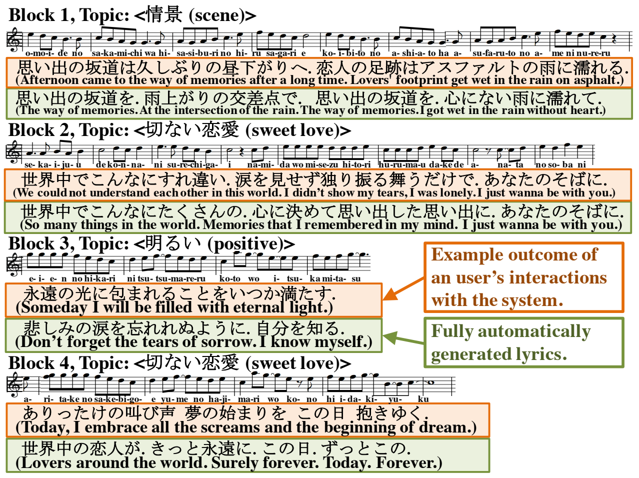
Previous method 2 (interaction without topic transition) We implemented another type of LyriSys that has a restricted topic transition function. This LyriSys calculates the simple tri-gram probability $P(w_i|w_{i-2}, w_{i-1}, s_i)$. The purpose of this task is to compare LyriSys with the previous interface,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

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

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 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 4 shows an example of lyrics that were created when a user used LyriSys (i.e., the user set the musical structure and the story, and selected or edited the recommended lyrics). Moreover, a fully automatically generated lyric is also shown in Figure 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 “涙を見せず (not show my tears)” and “あなたのそばに (with you)” were created when the topic was $\langle$sweet love$\rangle$. Note that these phrases are recommended by LyriSys.


Block 1, Topic: <情景 (scene)>
o-mo-i- de no sa-ka-mi-chi wa hi- sa-si-bu-ri no hi- ru sa-ga-ri e ko- i-bi-to no a- shi-a- to ha a- su-fa-ru-to no a- me ni nu-re-ru
Block 2, Topic: <切ない恋愛 (sweet love)>
se- ka- i- ju- u de ko-n- na- ni su-re-chi-ga- i na-mi- da wo mi-se-zu hi-to-ri hu-ru-ma-u da-ke de a- na- ta no so- ba ni
Block 3, Topic: <明るい (positive)>
e- i- e- n no hi-ka-ri ni tsu- tsu-ma-re-ru ko-to wo i- tsu- ka mi-ta- su
Example outcome of an user's interactions with the system.
Fully automatically generated lyrics.
Block 4, Topic: <切ない恋愛 (sweet love)>
a- ri- ta-ke no sa-ke-bi-go- e yu- me no ha-ji- ma-ri wo ko- no hi i- da- ki- yu- ku

思い出の坂道は久しぶりの昼下がりへ. 恋人の足跡はアスファルトの雨に濡れる.
(Afternoon came to the way of memories after a long time. Lovers' footprint get wet in the rain on asphalt.)

思い出の坂道を. 雨上がりの交差点で. 思い出の坂道を. 心にない雨に濡れて.
(The way of memories. At the intersection of the rain. The way of memories. I got wet in the rain without heart.)

世界中でこんなにすれ違い. 涙を見せず独り振る舞うだけで. あなたのそばに.
(We could not understand each other in this world. I didn't show my tears, I was lonely. I just wanna be with you.)

世界中でこんなにたくさんの. 心に決めて思い出した思い出に. あなたのそばに.
(So many things in the world. Memories that I remembered in my mind. I just wanna be with you.)

永遠の光に包まれることをいつか満たす.
(Someday I will be filled with eternal light.)

悲しみの涙を忘れれぬように. 自分を知る.
(Don't forget the tears of sorrow. I know myself.)

ありったけの叫び声 夢の始まりを この日 抱きゆく.
(Today, I embrace all the screams and the beginning of dream.)

世界中の恋人が. きっと永遠に. この日. ずっとこの.
(Lovers around the world. Surely forever. Today. Forever.)

Figure 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).

CONCLUSIONS AND FUTURE WORK

In this paper, 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-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 $\langle$scene$\rangle$, $\langle$dark$\rangle$ and $\langle$sweet love$\rangle$.

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.

ACKNOWLEDGMENTS

This study utilized the RWC Music Database (Popular Music). This work was supported by a Grant-in-Aid for JSPS Research Fellow Grant Number JP16J05945, JSPS KAKENHI Grant Numbers JP15K16045, and CREST, JST.

REFERENCES

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

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

3. Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. 2012. Markov Constraints for Generating Lyrics with Style. In *Proceedings of the 20th European Conference on Artificial Intelligence*. 115–120.

4. Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. 2002. RWC Music Database: Popular, Classical and Jazz Music Databases.. In *Proceedings of the 3rd of International Society for Music Information Retrieval*, Vol. 2. 287–288.

5. Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 230–237.

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

7. Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60.

8. Hugo Gonçalo Oliveira. 2015. Tra-la-Lyrics 2.0: Automatic Generation of Song Lyrics on a Semantic Domain. *Journal of Artificial General Intelligence* 6, 1 (2015), 87–110.

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

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

11. Rajeswari Sridhar, D Jalin Gladis, Kameswaran Ganga, and G Dhivya Prabha. 2014. Automatic Tamil lyric generation based on ontological interpretation for semantics. *Jornal of Sadhana* 39, 1 (2014), 97–121.

12. Tatsuji Ueda. 2010. *The writing lyrics textbook which is easy to understand (in Japanese)*. yamaha music media corporation.

13. Kento Watanabe, Yuichiroh Matsubayashi, Kentaro Inui, and Masataka Goto. 2014. Modeling Structural Topic Transitions for Automatic Lyrics Generation. In *Proceedings of The 28th Pacific Asia Conference on Language, Information and Computation*. 422–431.