LYRIC JUMPER: A LYRICS-BASED MUSIC EXPLORATORY WEB SERVICE BY MODELING LYRICS GENERATIVE PROCESS

Kosetsu Tsukuda **Keisuke Ishida** **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{k.tsukuda, ksuke-ishida, m.goto}@aist.go.jp

ABSTRACT

Each artist has their own taste for topics of lyrics such as "love" and "friendship." Considering such artist's taste brings new applications in music information retrieval: choosing an artist based on topics of lyrics and finding unfamiliar artists who have similar taste to a favorite artist. Although previous studies applied latent Dirichlet allocation (LDA) to lyrics to analyze topics, LDA was not able to capture the artist's taste. In this paper, we propose a topic model that can deal with the artist's taste for topics of lyrics. Our model assumes each artist has a topic distribution and a topic is assigned to each song according to the distribution. Our experimental results using a real-world dataset show that our model outperforms LDA in terms of the perplexity. By applying our model to estimate topics of 147,990 lyrics by 3,722 artists, we implement a web service called *Lyric Jumper* that enables users to explore lyrics based on the estimated topics. Lyric Jumper provides functions such as artist's topic taste visualization and topic-similarity-based artist recommendation. We also analyze operation logs obtained from 12,353 users on Lyric Jumper and show the usefulness of Lyric Jumper especially in recommending topic-related phrases in lyrics.

1. INTRODUCTION

Different artists have different tastes in lyrics. Some artists tend to sing about "love," while other artists tend to sing about "friendship." When listening to music, people choose artists according to not only musical audio content, such as music genre, mood, melody, vocal timbre, and rhythm, but also the topics of lyrics [2, 21]. However, the potential of using the topics of lyrics has not yet been fully exploited in the field of music information retrieval (MIR). For example, it is difficult to choose an artist based on the topics of their lyrics, find unfamiliar artists that are similar to the user's favorite artist in terms of the topics of the lyrics, and listen to a song that has the user's favorite topic of lyrics. The goal of this research is to achieve lyrics-based MIR that can leverage the topics of lyrics at both artist and song levels.

One approach for lyrics-based MIR is to directly use the words in lyrics. Users input some words as a query [5, 27] or can find the same phrase in the lyrics of another song while they are listening to music [9]. Another approach is to use a topic model because it can deal with the underlying meanings of lyrics. The topic is usually represented by a distribution over the vocabulary, and the meaning of topics (*e.g.*, "love" or "friendship") is determined based on the distribution. In lyrics-based MIR, it has been popular to use latent Dirichlet allocation (LDA) [1] as a topic model. LDA models each song as a mixture of topics and assigns a topic to each word in the song's lyrics. Since LDA does not take the set of songs of each artist into account, it is not able to capture the artist's taste for topics of lyrics.

In light of the above, we propose a topic model that considers the artist's taste for topics of lyrics. In the lyrics generative process of our model, each artist has a distribution over topics that reflects the artist's taste for topics in their lyrics. In addition, since it is common to decide the theme for a song before starting to write its lyrics [4, 36], our model assigns one topic to each song. That is, given a topic k assigned to a song, topic k is also assigned to the words in its lyrics. We also use the background word distribution because not all the words in lyrics are related to the topic.

By using our proposed model, we implemented a lyrics-based music exploratory web service, called *Lyric Jumper*[1 2]. Lyric Jumper aims to enable users to explore lyrics and enjoy music in a more flexible way by considering songs' topics. Our proposed model automatically assigns 1 of 20 topics for each song, where the 20 topics are also automatically estimated by our model. Lyric Jumper provides several topic-based functions such as visualization of the topic tendency for a given artist, artist ranking based on topics, and artist recommendation based on the topic distribution similarity.

Our main contributions in this paper are as follows.

- To the best of our knowledge, this is the first study modeling a lyrics generative process by considering the artist's taste for topics of lyrics and assuming each song has one topic. (Section 3)
- We quantitatively evaluated our model by using a real-world song dataset provided by a lyrics distribution company. Our experimental results show that our proposed model outperformed the conventional LDA in terms of the perplexity. (Section 4)

[1] https://lyric-jumper.petitlyrics.com
[2] The demonstration video: https://youtu.be/5V9kHnelSAk

- By using our proposed model, we implemented a web service, called *Lyric Jumper*, that enables users to search for songs based on the topics of their lyrics. We also analyzed the search logs obtained from more than 12,000 users and showed the impact of Lyric Jumper on users' search behavior. (Section 5)

2. RELATED WORK

Previous studies have used lyrics for various objectives such as lyrics-to-audio alignment [7, 22, 35], analyzing lyrics characteristics [8, 13, 16, 29, 34], accurately finding lyrics [11, 19, 24], genre or mood classification [15, 25, 26, 38–40], songwriting support [30], and video generation [10]. This section describes more related studies in terms of (1) lyrics-based music retrieval/browsing systems and (2) topic-based lyrics analysis and applications.

2.1 Lyrics-Based Music Retrieval/Browsing Systems

Brochu and de Freitas [5] modeled music and text jointly so that users can search song databases using music and/or text as input. Müller *et al.* [27] automatically annotated audio recordings of a given song with its corresponding lyrics and realized a query-by-lyrics retrieval system. When a user selects a query result, the system can directly navigate to the corresponding matching positions within the audio. Detecting songs from the user's singing lyrics is also a popular research topic [14,37]. Fujihara *et al.* [9] proposed the concept of a "Music Web" where songs were hyperlinked to each other based on the phrases of lyrics. This enables users to jump to the same phrase in the lyrics of another song by clicking a linked phrase while they are listening to music. Visualization is also a useful approach to browse a music collection. *SongWords* [3], which is an application for tabletop computers, displays a music collection on a two-dimensional canvas based on self-organizing maps for lyrics and tags. *Lyricon* [23] is a system that automatically selects and displays icons that match the word sequences of lyrics so that users can intuitively understand the lyrics.

Although these studies directly use the words in lyrics, we consider topics that are automatically estimated from lyrics. Our approach has an advantage in that users can explore lyrics based on the underlying meanings of the lyrics.

2.2 Topic-Based Lyrics Analysis and Applications

Since a topic model can learn the underlying meanings of lyrics, it has been used in various studies, including lyrics analysis [17, 31, 33], lyrics retrieval applications [32], and a music player [28]. In terms of lyrics analysis, Sharma and Murty [33] analyzed the hidden sentimental structure behind lyrics by using LDA and revealed that some of the detected topics correspond to sentiments. Similarly, by applying LDA to rap lyrics, not only expected topics such as "street life" and "religion" but also unexpected ones such as "family/childhood" can be discovered [17]. Ren *et al.* [31] tackled the problem of predicting the popularity of a music track by considering lyrics topics and found that more than half of the popular tracks are related to the topic of "love." Regarding applications, *LyricsRadar* [32] is a lyrics retrieval system that visualizes the topic ratio for each song by using the topic radar chart and enables users to find their favorite lyrics interactively. Nakano and Goto [28] presented a music playback interface *LyricList-Player* that enables users to see word sequences of other songs similar to the sequence currently being played back, where the similarity is computed based on the topic.

In these studies, LDA is used as a topic model, where it is assumed that each song has a topic distribution and a topic is assigned to each word in the lyrics. We propose a new topic model that assumes each artist has a topic distribution and a topic is assigned to each song. Since our model outperforms LDA (see Section 4), there is the potential for improving previous studies on lyrics analysis and applications by using our model.

The study closest to ours is that of Kleedorfer *et al.* [18], who applied non-negative matrix factorization to lyrics for clustering them and manually labeled the cluster names. Our study differs from theirs in that we consider the artist's taste for topics of lyrics, and this enables users to find their favorite lyrics based on the relationships between artists and topics. Moreover, we not only propose a new model but also implement a web service so that everyone can explore lyrics with a real world dataset.

3. MODEL AND INFERENCE

In this section, after summarizing the notations used in our model in Section 3.1, we first describe LDA in Section 3.2 and then propose our model in Section 3.3.

3.1 Notations

Given a lyrics dataset, let A be the set of artists in the dataset. Let R_a be the number of songs of artist $a \in A$ in the dataset; then the set of a's songs is given by $\{S_{ar}\}_{r=1}^{R_a}$, where S_{ar} represents the rth song of a. Moreover, let V_{ar} be the number of words in the lyrics of S_{ar}; then S_{ar} can be represented by $S_{ar} = \{v_{arj}\}_{j=1}^{V_{ar}}$, where v_{arj} is the jth word in S_{ar}. Hence, the set of words of all artists' lyrics is given by $D = \{\{\{v_{arj}\}_{j=1}^{V_{ar}}\}_{r=1}^{R_a}\}_{a \in A}$.

3.2 LDA (Latent Dirichlet Allocation)

When LDA is used as a generative process of lyrics, it is assumed that (1) each song has a distribution over topics, (2) a topic is assigned to each word in the song's lyrics according to the distribution, and (3) a word is generated from the topic's distribution over words. Figure 1(a) shows the graphical model of LDA, where the shaded and unshaded circles represent the observed and unobserved variables, respectively. In the figure, K is the number of topics, θ is the song-topic distribution, and ϕ is the topic-word distribution. We assume that θ and ϕ have Dirichlet priors of α and β, respectively. The generative process of LDA is described in Algorithm 1.

3.3 Artist's Taste (AT) Model

Although previous studies reported the usefulness of applying LDA to lyrics [17, 28, 31–33], LDA does not take artist information into account in the generative process.

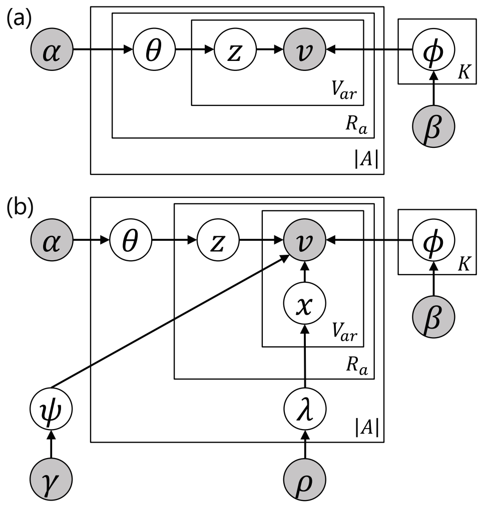

Figure 1. Graphical models of (a) baseline LDA and (b) proposed artist's taste (AT) model.

It is reasonable to assume that each artist has their own taste for topics of lyrics. For example, one artist may tend to sing lyrics related to the topic of "love," while another artist may tend to sing lyrics related to the topic of "life."

In light of the above, we propose a model that considers the *artist's taste* for topics. Figure 1(b) shows the graphical model of our proposed model. In our model, each artist has a distribution over topics (θ). When people write lyrics, the writer typically decides the theme (*i.e.*, the topic) before starting to write the lyrics [4, 36]. Hence, we assume each song has a topic z that is generated from θ. However, not all of the words in the lyrics are related to the topic. For example, although "thing" and "this" frequently appear in many lyrics, usually these words do not represent a specific topic. To solve this problem, we use the idea of background words [6]. In Figure 1(b), ψ represents the background word distribution, where words that are not related to any topic have high occurrence probabilities. Each artist has a Bernoulli distribution λ that controls the weights of influence for a song topic and background words. To be more specific, when artist a chooses a word in a song, we assume that the choice is influenced by the song topic with probability λ_{a0} ($x = 0$) and by background words with probability λ_{a1} ($x = 1$), where $\lambda_{a0} + \lambda_{a1} = 1$. When $x = 0$, a word is generated from the topic's distribution over words, while when $x = 1$, a word is generated from the background word distribution ψ. The generative process of the AT model is described in Algorithm 2.

3.4 Inference

To learn the parameters of our proposed model, we use collapsed Gibbs sampling [12] to obtain samples of hidden variable assignment. Since we use a Dirichlet prior for θ, ϕ, and ψ and a Beta prior for λ, we can analytically calculate the marginalization over the parameters. The marginalized joint distribution of D, latent variables $Z = \{\{z_{ar}\}_{r=1}^{R_a}\}_{a \in A}$, and latent variables $X = \{\{\{x_{arj}\}_{j=1}^{V_{ar}}\}_{r=1}^{R_a}\}_{a \in A}$ is computed as follows:

Algorithm 1 LDA generative process

for each topic $k \in \{1, \cdots, K\}$ **do**
  Draw $\phi_k \sim \mathit{Dirichlet}(\beta)$
end for
for each artist a in A **do**
  for each song S_{ar} **do**
    Draw $\theta_{ar} \sim \mathit{Dirichlet}(\alpha)$
    for each word v_{arj} in S_{ar} **do**
      Draw a topic $z_{arj} \sim \mathit{Multinomial}(\theta_{ar})$
      Draw a word $v_{arj} \sim \mathit{Multinomial}(\phi_{z_{arj}})$
    end for
  end for
end for

Algorithm 2 AT model generative process

for each topic $k \in \{1, \cdots, K\}$ **do**
  Draw $\phi_k \sim \mathit{Dirichlet}(\beta)$
end for
Draw $\psi \sim \mathit{Dirichlet}(\gamma)$
for each artist a in A **do**
  Draw $\theta_a \sim \mathit{Dirichlet}(\alpha)$
  Draw $\lambda_a \sim \mathit{Beta}(\rho)$.
  for each song S_{ar} **do**
    Draw a topic $z_{ar} \sim \mathit{Multinomial}(\theta_a)$
    for each word v_{arj} in S_{ar} **do**
      Draw switch $x \sim \mathit{Bernoulli}(\lambda_a)$
      if $x = 0$ **then**
        Draw a word $v_{arj} \sim \mathit{Multinomial}(\phi_{z_{ar}})$
      else if $x = 1$ **then**
        Draw a word $v_{arj} \sim \mathit{Multinomial}(\psi)$
      end if
    end for
  end for
end for

$$\begin{aligned} &P(D, Z, X|\alpha, \beta, \gamma, \rho) \\ &= \iiiint P(D, Z, X|\mathbf{\Theta}, \mathbf{\Phi}, \psi, \mathbf{\Lambda})P(\mathbf{\Theta}|\alpha) \\ &\quad \times P(\mathbf{\Phi}|\beta)P(\psi|\gamma)P(\mathbf{\Lambda}|\rho)d\mathbf{\Theta}d\mathbf{\Phi}d\psi d\mathbf{\Lambda}, \end{aligned} \tag{1}$$

where $\mathbf{\Theta} = \{\theta_a\}_{a \in A}$, $\mathbf{\Phi} = \{\phi_k\}_{k=1}^{K}$, and $\mathbf{\Lambda} = \{\lambda_a\}_{a \in A}$. By integrating out those parameters, we can compute Equation (1) as follows:

$$\begin{aligned} &P(D, Z, X|\alpha, \beta, \gamma, \rho) \\ &\propto \prod_{a \in A} \frac{\Gamma(\rho + N_{a0})\Gamma(\rho + N_{a1})}{\Gamma(2\rho + N_a)} \frac{\prod_{v \in V} \Gamma(N_{1v} + \gamma)}{\Gamma(N_1 + \gamma|V|)} \\ &\times \prod_{k=1}^{K} \frac{\prod_{v \in V} \Gamma(N_{kv} + \beta)}{\Gamma(N_k + \beta|V|)} \prod_{a \in A} \frac{\prod_{k=1}^{K} \Gamma(R_{ak} + \alpha)}{\Gamma(R_a + \alpha K)}. \end{aligned} \tag{2}$$

Here, N_{a0} and N_{a1} are the number of words in a's songs such that $x = 0$ and $x = 1$, respectively, and $N_a = N_{a0} + N_{a1}$. The term N_{1v} represents the number of times that word v was chosen under the condition of $x = 1$, and $N_1 = \sum_{v \in V} N_{1v}$ where V is the set of unique words in D. Furthermore, $N_k = \sum_{v \in V} N_{kv}$ where N_{kv} is the number of times word v is assigned to topic k under the condition of $x = 0$. Finally, R_{ak} is the number of times topic k is assigned to a's song, and $R_a = \sum_{k=1}^{K} R_{ak}$.

For the Gibbs sampler, given the current state of all but one variable z_{ar}, the new latent assignment of z_{ar} is sampled from the following probability:

$$P(z_{ar} = k | D, X, Z_{\backslash ar}, \alpha, \beta, \gamma, \rho)$$
$$\propto \frac{R_{ak\backslash ar} + \alpha}{R_a - 1 + \alpha K} \frac{\Gamma(N_{k\backslash ar} + \beta|V|)}{\Gamma(N_{k\backslash ar} + N_{ar} + \beta|V|)}$$
$$\times \prod_{v \in V} \frac{\Gamma(N_{kv\backslash ar} + N_{arv} + \beta)}{\Gamma(N_{kv\backslash ar} + \beta)}, \quad (3)$$

where $\backslash ar$ represents the procedure excluding the rth song of a. Moreover, N_{ar} and N_{arv} represent the number of words in the rth song of a and the number of times word v appears in the rth song of a, respectively.

In addition, given the current state of all but one variable x_{arj}, the probability at which $x_{arj} = 0$ is given by:

$$P(x_{arj} = 0 | D, X_{\backslash arj}, Z, \alpha, \beta, \gamma, \rho)$$
$$\propto \frac{\rho + N_{a0\backslash arj}}{2\rho + N_a - 1} \frac{N_{z_{ar} v_{arj} \backslash arj} + \beta}{N_{z_{ar}\backslash arj} + \beta|V|}, \quad (4)$$

where $\backslash arj$ represents the procedure excluding the jth word in the rth song of a. Similarly, the probability at which $x_{arj} = 1$ is computed as follows:

$$P(x_{arj} = 1 | D, X_{\backslash arj}, Z, \alpha, \beta, \gamma, \rho)$$
$$\propto \frac{\rho + N_{a1\backslash arj}}{2\rho + N_a - 1} \frac{N_{1 v_{arj} \backslash arj} + \gamma}{N_{1\backslash arj} + \gamma|V|}. \quad (5)$$

Finally, we can make the point estimates of the integrated out parameters as follows:

$$\theta_{ak} = \frac{R_{ak} + \alpha}{R_a + \alpha K}, \quad \phi_{kv} = \frac{N_{kv} + \beta}{N_k + \beta|V|}, \quad \psi_v = \frac{N_{1v} + \gamma}{N_1 + \gamma|V|},$$
$$\lambda_{a0} = \frac{N_{a0} + \rho}{N_a + 2\rho}, \quad \lambda_{a1} = \frac{N_{a1} + \rho}{N_a + 2\rho}. \quad (6)$$

4. EVALUATION

In this section, we carry out a quantitative evaluation to answer the following research question: is adopting the artist's taste for topics effective to model the lyrics generative process?

[Dataset] We used the lyrics of commercially available popular music. Those lyrics with the song's title and artist name were provided by one of the largest companies for commercial lyrics distribution. We collected data on the top 1,000 artists in terms of the number of lyrics that are available as of the end of December 2016; this gave us 93,716 songs in total. We then extracted Japanese nouns from each song's lyrics by using MeCab [20], which is a Japanese morphological analyzer. Nouns that appeared in less than 10 lyrics were eliminated. Although our proposed model is language-independent, we used only Japanese words because of the understandability of the estimated topics for Japanese users of Lyric Jumper that we will describe in Section 5. From each of lyrics, we randomly sampled 80% of the nouns for training data and used the remaining 20 % for test data.

[Settings] In terms of hyperparameters, in line with other topic modeling work, we set $\alpha = \frac{1}{K}$ and $\beta = \frac{50}{|V|}$ in LDA and the artist's taste (AT) model. In addition, in the AT model, we set $\gamma = \frac{50}{|V|}$ and $\rho = 0.5$. To compare the performance of LDA and the AT model, we use the perplexities of the two models. Perplexity is widely used to compare the performance of statistical models [1], and the lower value represents the better performance. In terms of the number of topics, we compute the perplexity for $K = 2, 4, 6, 8, 10, 20, 30, 40$, and 50.

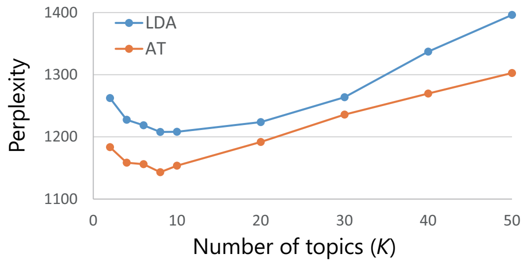

Figure 2. Perplexity for baseline LDA and proposed AT model (the lower, the better).

[Results] Figure 2 shows the perplexity. As can be seen, regardless of the number of topics, the AT model outperforms LDA. In both methods, the perplexity reaches a minimum when the number of topics is eight. The difference of perplexity between the two models becomes larger as the number of topics increases. From these results, we can conclude that the AT model is superior to LDA for modeling a lyrics generative process and confirmed the effectiveness of modeling a topic distribution for each artist.

5. LYRIC JUMPER

By using the AT model, we implemented a lyrics-based music exploratory web service called *Lyric Jumper* that anyone can use for free without registration. In this section, we describe the implementation and functions of Lyric Jumper followed by the log analysis based on the users' operation logs obtained from the web service.

5.1 Implementation

For Lyric Jumper, the lyrics are provided by the aforementioned company for commercial lyrics distribution. We used all the lyrics that are available as of the end of December 2016 and extracted Japanese nouns. To guarantee the topic quality, we eliminated artists who had < 10 songs and nouns that appeared in < 10 songs. This gave us 147,990 songs by 3,722 artists.

As for the number of topics K, if K is too small, users would soon get bored of using Lyric Jumper, while if K is too large, it would be difficult to understand the difference between topics since many similar topics are generated. Hence, after comparing the topic qualities for several K values, we set $K = 20$ for Lyric Jumper, although $K = 8$ achieved the best result in terms of the perplexity in Section 4. After automatically estimating the 20 topics by using the AT model, we manually labeled topic names so that users can easily understand the characteristics of each topic. Examples of topic names are "life," "sentimental," and "adolescence." Although five topics are related to "love," our model was able to distinguish between sub-

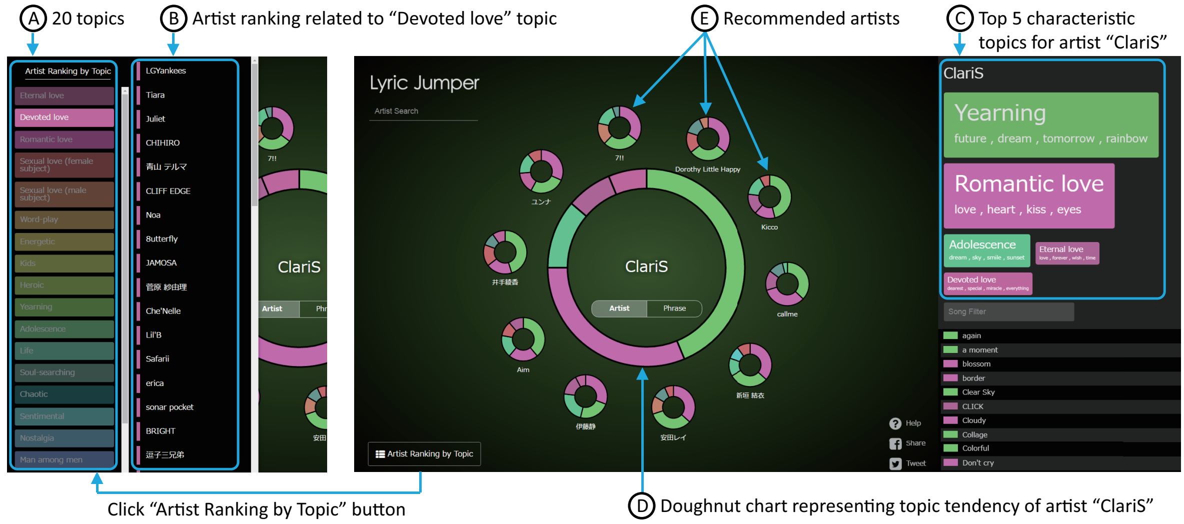

Figure 3. Overview of Lyric Jumper.

tle differences of love: "eternal love," "devoted love," "romantic love," "sexual love (female subject)," and "sexual love (male subject)."

5.2 Function

Lyric Jumper mainly provides six functions. The following sections describe the functions one by one.

5.2.1 *Artist Ranking*

Lyric Jumper displays 20 topic names as shown in Figure 3 Ⓐ. By clicking one of the 20 topics, Lyric Jumper shows up to 100 artists related to the topic (Figure 3 Ⓑ). This enables the user to see many artists related to the topic of his/her interest. The user can also find that unexpected artists are related to the topic.

Intuitively, given a topic k, artists are ranked based on both their topic ratio of k (θ_{ak}) and the number of songs assigned to k (R_{ak}) so that artists more closely related to the topic are ranked higher. To be more specific, we sort all the artists in A by using the rank of topic k in θ_a as the first key (the smaller the better) and using the number of songs assigned to k (R_{ak}) as the second key (the larger the better). Note that artists whose rank of k in θ_a is lower than five are not included in the ranking because Lyric Jumper shows the top five topics for each artist as we will describe in Section 5.2.2. Finally, we select the top 100 artists in the sorted list and show them to the users.

5.2.2 *Topic Tendency Visualization*

When a user clicks an artist, Lyric Jumper visualizes the topic tendency of the artist. In this function, given artist a, the top five topics in terms of the occurrence probability in θ_a are displayed in rectangles (Figure 3 Ⓒ). The size of a rectangle corresponds to the topic probability: the larger it is, the higher the probability is. With this function, a user can not only understand the topic tendency of the artist's lyrics but also find out that the artist sings songs with unexpected topics. We also manually selected four characteristic words for each topic and displayed them below the topic name so that users can more easily understand the meaning of the topic. In addition, Lyric Jumper visualizes the topic tendency using a doughnut chart where the circle is divided according to the ratio of the top five topics (Figure 3 Ⓓ).

5.2.3 *Artist Recommendation*

Since similar artists are one of the important information needs in MIR [21], Lyric Jumper provides a similar artists recommendation function. Lyric Jumper recommends 10 artists in terms of the topic similarity (Figure 3 Ⓔ). Among the 10 artists, eight artists are popular and two artists are minor. By displaying minor artists as well as popular ones, Lyric Jumper aims to encourage the user to listen to unfamiliar artists' songs that are related to his/her favorite artist by the topic similarity. By clicking a recommended artist's graph, the user can jump to the artist's search result.

Given a selected artist a, we compute the similarity between a and each artist $a' \in A \setminus \{a\}$ based on Jensen-Shannon divergence (JSD) between θ_a and $\theta_{a'}$. The smaller the JSD value is, the higher the similarity between artists is. After computing the similarities, we select the top eight similar artists who have $\geq m$ songs in the dataset (*i.e.*, popular artists) and the top two artists who have $< m$ songs (*i.e.*, minor artists) and show those 10 artists to the users. On Lyric Jumper, m is set to 100.

5.2.4 *Phrase Emphasized Lyrics Visualization*

When a user clicks a topic of the selected artist, Lyric Jumper shows the list of song titles of the artist that are assigned to the topic (the song ranking method will be described in Section 5.2.5). When the user clicks a title in the list, the song's lyrics are displayed (Figure 4). In the lyrics, lines[3] related to the topic are displayed with emphasis: the stronger the relation is, the larger the font size becomes and the darker the color becomes. This enables a user to easily understand the characteristics of the lyrics such as "the latter half of the lyrics is strongly related to the topic." Users can also watch the song's videos on Lyric Jumper by clicking the "YouTube Search" button. Lyric Jumper shows the search results obtained from YouTube[4]

[3] We use the terms "phrase" and "line" interchangeably.
[4] https://www.youtube.com/

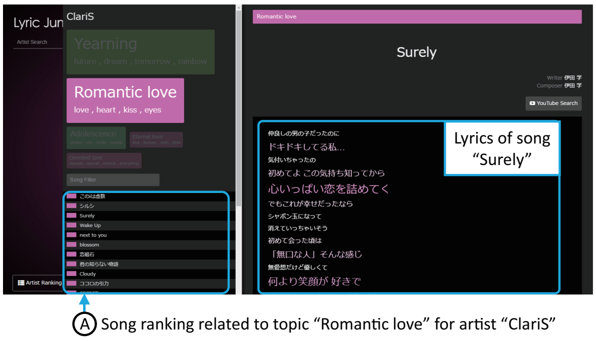

Figure 4. Phrase emphasized lyrics visualization.

where the query is the artist name and the song title.

The relevance score between a line in the lyrics and the topic is computed as follows. We first assign scores for the nouns in ϕ_k. Let $rank(k, v)$ be the occurrence probability rank of noun v in ϕ_k. The score of v is given by $w\_rel(k, v) = 101 - rank(k, v)$ if $rank(k, v) \leq 100$ and $w\_rel(k, v) = 0$ otherwise. Line l consists of $n \geq 0$ nouns and can be represented by $l = (v_1, \cdots, v_n)$. The relevance score of l with topic k is given by $l\_rel(k, l) = \sum_{i=1}^{n} w\_rel(k, v_i)$. After computing the scores of all lines in a song's lyrics, the scores are normalized to fit into the interval $[0, 1]$ by min-max normalization. The font size linearly changes from 16 pt for a score of 0 to 36 pt for a score of 1; the color density also linearly changes from #FFFFFF for a score of 0 to the topic color for a score of 1.

5.2.5 *Artist's Songs Ranking*

As mentioned in Section 5.2.4, when a user clicks a topic of the selected artist, Lyric Jumper returns the ranked list of songs in the topic (Figure 4 Ⓐ). Songs are sorted in descending order of relevance to the topic so that the user can easily access songs that are strongly related to the topic.

The relevance score between song s and topic k is given by $s\_rel(k, s) = \frac{1}{|L_s|} \sum_{l \in L_s} l\_rel(k, l)$, where L_s is the set of lines in s's lyrics. That is, we assume that the relatedness between s and k can be represented by the average relevance between k and each line in s's lyrics.

5.2.6 *Phrase Recommendation*

By clicking the “Phrase” button after selecting an artist's topic, Lyric Jumper recommends phrases related to the topic in the artist's songs (Figure 5). Moreover, every time the user clicks the “PUSH!” button, a new phrase is recommended. This function enables users to understand there are various expressions to deliver messages about the topic. When the user clicks a phrase, Lyric Jumper shows the corresponding lyrics in the same way as in Section 5.2.4.

Given artist a and topic k, the recommended phrases are selected as follows. In the ith round ($i = 1, 2, \cdots$), we pool lines that have the ith highest score of $l\_rel(k, l)$ from a's songs in order of decreasing $s\_rel(k, s)$. This round is repeated until the number of pooled lines is equal to 100. Lyric Jumper recommends phrases from the pooled list in random order so that users can see different phrases every time the user accesses Lyric Jumper.

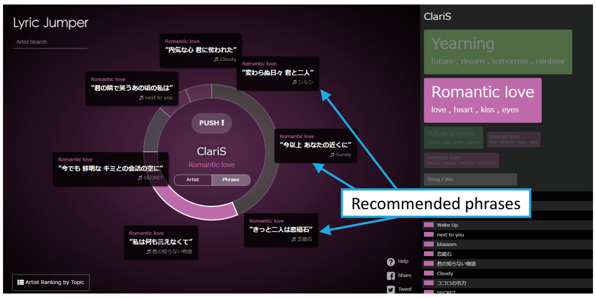

Figure 5. Phrase recommendation.

Function	PC	Smartphone
Artist ranking	2,092	30,295
Artist recommendation	1,706	4,016
Artist's songs ranking	5,399	14,665
Phrase recommendation	4,997	253,430

Table 1. Statistics of use frequency of each function.

5.3 Log Analysis

We released Lyric Jumper as a web service open to the public on 2/21/2017. To analyze users' exploratory behavior on Lyric Jumper, we obtained operation logs for 30 days (2/21 to 3/22). The numbers of unique PC users and smartphone users are 1,288 and 11,065, respectively. The use frequencies of each function are summarized in Table 1. We can see that the use frequencies of the artist ranking and artist's songs ranking are high. These results indicate that exploratory search for artists and songs based on topics can stimulate the user's interest. It can also be observed that for smartphone users in particular, the phrase recommendation function was used frequently: the push button was clicked as many as 253,430 times. This data shows the user's high information needs regarding finding lyrics using phrases related to a topic. Compared to these functions, the recommended artists were not clicked very often. To encourage the artist-similarity-based lyrics exploratory search, a more sophisticated interface for the recommendation deserves to be explored; we leave this as future work.

6. CONCLUSION

In this paper we proposed a topic model that incorporates the artist's taste for topics of lyrics. Our experimental results showed that our model outperformed the state-of-the-art LDA model regardless of the number of topics in terms of the perplexity. We also released a lyrics-based music exploratory web service called Lyric Jumper, where we applied our model to 147,990 lyrics by 3,722 artists. Our log analysis results show that the phrase recommendation function, which recommends phrases from the lyrics of the artist's songs related to the selected topic, achieved a particularly high use frequency.

For future work, since our model is language-independent, we plan to apply our model to English lyrics and implement an English version of Lyric Jumper. We are also interested in combining topics obtained by our model with other features such as audio content and tags. This would enable users to explore songs by adapting their search intent with increased flexibility.

7. ACKNOWLEDGMENTS

The authors would like to extend their appreciation to SyncPower Corporation for providing lyrics data. This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

8. REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[2] D. Bainbridge, S.J. Cunningham, and J.S. Downie. How people describe their music information needs: A grounded theory analysis of music queries. In *Proceedings of the 4th International Conference on Music Information Retrieval*, ISMIR'03, pages 221–222, 2003.

[3] D. Baur, B. Steinmayr, and A. Butz. SongWords: Exploring music collections through lyrics. In *Proceedings of the 11th International Conference on Music Information Retrieval*, ISMIR'10, pages 531–536, 2010.

[4] M. Baxter. Voices of resistance, voices of transcendence: Musicians as models of the poetic - political imagination. *International Journal of Education & the Arts*, 11:1–24, 2010.

[5] E. Brochu and N. de Freitas. "Name that song!" a probabilistic approach to querying on music and text. In *Proceedings of the 15th International Conference on Neural Information Processing Systems*, NIPS'02, pages 1505–1512. 2002.

[6] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, pages 241–248, 2006.

[7] G. Dzhambazov, A. Srinivasamurthy, S. Sentürk, and X. Serra. On the use of note onsets for improved lyrics-to-audio alignment in Turkish makam music. In *Proceedings of the 17th International Conference on Music Information Retrieval*, ISMIR'16, pages 716–722, 2016.

[8] R. J. Ellis, Z. Xing, J. Fang, and Y. Wang. Quantifying lexical novelty in song lyrics. In *Proceedings of the 16th International Conference on Music Information Retrieval*, ISMIR'15, pages 694–700, 2015.

[9] H. Fujihara, M. Goto, and J. Ogata. Hyperlinking lyrics: A method for creating hyperlinks between phrases in song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval*, ISMIR'08, pages 281–286, 2008.

[10] S. Funasawa, H. Ishizaki, K. Hoashi, Y. Takishima, and J. Katto. Automated music slideshow generation using web images based on lyrics. In *Proceedings of the 11th International Conference on Music Information Retrieval*, ISMIR'10, pages 63–68, 2010.

[11] G. Geleijnse and J. H. M. Korst. Efficient lyrics extraction from the web. In *Proceedings of the 7th International Conference on Music Information Retrieval*, ISMIR'06, pages 371–372, 2006.

[12] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

[13] H. Hirjee and D. G. Brown. Automatic detection of internal and imperfect rhymes in rap lyrics. In *Proceedings of the 10th International Conference on Music Information Retrieval*, ISMIR'09, pages 711–716, 2009.

[14] T. Hosoya, M. Suzuki, A. Ito, and S. Makino. Lyrics recognition from a singing voice based on finite state automaton for music information retrieval. In *Proceedings of the 6th International Conference on Music Information Retrieval*, ISMIR'05, pages 532–535, 2005.

[15] X. Hu and J. S. Downie. When lyrics outperform audio for music mood classification: A feature analysis. In *Proceedings of the 11th International Conference on Music Information Retrieval*, ISMIR'10, pages 619–624, 2010.

[16] X. Hu and B. Yu. Exploring the relationship between mood and creativity in rock lyrics. In *Proceedings of the 12th International Conference on Music Information Retrieval*, ISMIR'11, pages 789–794, 2011.

[17] C. Johnson-Roberson and M. Johnson-Roberson. Temporal and regional variation in rap lyrics. In *NIPS Workshop on Topic Models: Computation, Application and Evaluation*, NIPSW'13, 2013.

[18] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval*, ISMIR'08, pages 287–292, 2008.

[19] P. Knees, M. Schedl, and G. Widmer. Multiple lyrics alignment: Automatic retrieval of song lyrics. In *Proceedings of the 6th International Conference on Music Information Retrieval*, ISMIR'05, pages 564–569, 2005.

[20] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP'04, pages 230–237, 2004.

[21] J. H. Lee and J. S. Downie. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. In *Proceedings of the 5th International Conference on Music Information Retrieval*, ISMIR'04, pages 989–992, 2004.

[22] K. Lee and M. Cremer. Segmentation-based lyrics-audio alignment using dynamic programming. In *Proceedings of the 9th International Conference on Music Information Retrieval*, ISMIR'08, pages 395–400, 2008.

[23] W. Machida and T. Itoh. Lyricon: A visual music selection interface featuring multiple icons. In *Proceedings of the 15th International Conference on Information Visualisation*, IV'11, pages 145–150, 2011.

[24] R. Macrae and S. Dixon. Ranking lyrics for online search. In *Proceedings of the 13th International Conference on Music Information Retrieval*, ISMIR'12, pages 361–366, 2012.

[25] R. Mayer, R. Neumayer, and A. Rauber. Rhyme and style features for musical genre classification by song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval*, ISMIR'08, pages 337–342, 2008.

[26] R. Mayer and A. Rauber. Music genre classification by ensembles of audio and lyrics features. In *Proceedings of the 12th International Conference on Music Information Retrieval*, ISMIR'11, pages 675–680, 2011.

[27] M. Müller, F. Kurth, D. Damm, C. Fremerey, and M. Clausen. Lyrics-based audio retrieval and multimodal navigation in music collections. In *Proceedings of the 11th European Conference on Digital Libraries*, ECDL'07, pages 112–123, 2007.

[28] T. Nakano and M. Goto. LyricListPlayer: A consecutive-query-by-playback interface for retrieving similar word sequences from different song lyrics. In *Proceedings of the Sound and Music Computing Conference 2016*, SMC'16, pages 344–349, 2016.

[29] E. Nichols, D. Morris, S. Basu, and C. Raphael. Relationships between lyrics and melody in popular music. In *Proceedings of the 10th International Conference on Music Information Retrieval*, ISMIR'09, pages 471–476, 2009.

[30] T. O'Hara, N. Schüler, Y. Lu, and D. Tamir. Inferring chord sequence meanings via lyrics: Process and evaluation. In *Proceedings of the 13th International Conference on Music Information Retrieval*, ISMIR'12, pages 463–468, 2012.

[31] J. Ren, J. Shen, and R. J. Kauffman. What makes a music track popular in online social networks? In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW'16 Companion, pages 95–96, 2016.

[32] S. Sasaki, K. Yoshii, T. Nakano, M. Goto, and S. Morishima. LyricsRadar: A lyrics retrieval system based on latent topics of lyrics. In *Proceedings of the 15th International Conference on Music Information Retrieval*, ISMIR'14, pages 585–590, 2014.

[33] G. Sharma and M. N. Murty. Mining sentiments from songs using latent Dirichlet allocation. In *Proceedings of the 10th International Conference on Advances in Intelligent Data Analysis X*, IDA'11, pages 328–339, 2011.

[34] A. Singhi and D. G. Brown. Are poetry and lyrics all that different? In *Proceedings of the 15th International Conference on Music Information Retrieval*, ISMIR'14, pages 471–476, 2014.

[35] V. Thomas, C. Fremerey, D. Damm, and M. Clausen. Slave: A score-lyrics-audio-video-explorer. In *Proceedings of the 10th International Conference on Music Information Retrieval*, ISMIR'09, pages 717–722, 2009.

[36] J. M. Toivanen, H. Toivonen, and A. Valitutti. Automatical composition of lyrical songs. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCC'13, pages 87–91, 2013.

[37] C. Wang, J. R. Jang, and W. Wang. An improved query by singing/humming system using melody and lyrics information. In *Proceedings of the 11th International Conference on Music Information Retrieval*, ISMIR'10, pages 45–50, 2010.

[38] X. Wang, X. Chen, D. Yang, and Y. Wu. Music emotion classification of chinese songs based on lyrics using TF*IDF and rhyme. In *Proceedings of the 12th International Conference on Music Information Retrieval*, ISMIR'11, pages 765–770, 2011.

[39] B. Wei, C. Zhang, and M. Ogihara. Keyword generation for lyrics. In *Proceedings of the 8th International Conference on Music Information Retrieval*, ISMIR'07, pages 121–122, 2007.

[40] M. Zaanen and P. Kanters. Automatic mood classification using TF*IDF based on lyrics. In *Proceedings of the 11th International Conference on Music Information Retrieval*, ISMIR'10, pages 75–80, 2010.