LyricListPlayer: 歌詞でリアルタイムにザッピングできる音楽再生インタフェース

中野 倫靖　　後藤 真孝 *

概要.　本論文では、音楽の再生に同期して、再生中の歌詞に局所的に類似している他の楽曲の歌詞を、リアルタイムに一覧表示して可視化する音楽再生インタフェース LyricListPlayer を提案する。一覧表示された類似歌詞をクリックすると、その該当箇所から再生できるザッピング機能も持つ。従来、歌詞の特徴的な単語をハイパーリンクさせるシステムや歌詞の類似度に基づいた検索インタフェース、音楽と歌詞の自動同期に基づく音楽再生インタフェースや Kinetic Typography 生成を支援するインタフェースが提案されてきたが、LyricListPlayer は歌詞に基づいたリアルタイムで局所的な音楽再生・音楽検索を実現する点で新しい。本インタフェースでは、歌詞の局所類似度を形態素単位で計算し、N 個の連続した形態素が似ている他の曲の歌詞を一覧表示する（$N = 1$〜5）。類似度計算において、潜在的ディリクレ配分法（LDA）で得られた各形態素のトピック分布（負担率）を用いることで、全ての形態素間の類似度計算を可能とした。

1　はじめに

ポピュラー音楽などの歌唱を伴う音楽において、歌詞は歌声（ボーカル）パート特有の要素である。歌詞があることで、音響的な観点からの鑑賞（メロディや音色、歌手の声質など）に加えて、意味的な観点からも鑑賞（情景や感情、ストーリーなど）を可能にする。また、歌詞は音楽聴取における感情移入において重要であり [21]、曲の聴取印象や記憶に影響を及ぼす [8, 19, 22] ことが知られている。したがって、歌詞の内容を意識することは、音楽鑑賞という体験をより豊かにするといえる。

そこで本研究では、歌詞を意識した音楽鑑賞を支援するため、鑑賞と検索の二つの観点に着目する。

【鑑賞】 歌声に同期した歌詞テキストの表示

【検索】 歌詞の内容に基づいた音楽検索

前者は楽曲中の歌詞のどこが再生されているかを意識しやすくなり、後者は楽曲間の類似性を歌詞の観点から意識しやすくなる。

ここで、歌詞を鑑賞しながら検索したり、検索しながら鑑賞したりするシームレスなインタラクションを導入することで、ストーリーに基づいて時々刻々と意味が変化する歌詞 [18] について、その時々で似ている歌詞を検索することを目的とする。これによって「今聴いている歌詞」やその意味が、他の曲ではどういう文脈で使われ、歌手によってどう歌われるのか等といった、楽曲を横断した観点での新しい音楽鑑賞を可能とする。またこれには、歌詞の単語に基づいた音楽検索において、検索クエリを考える代わりに、鑑賞によって見つけ出す楽しみもある。

Copyright is held by the author(s).

* 産業技術総合研究所

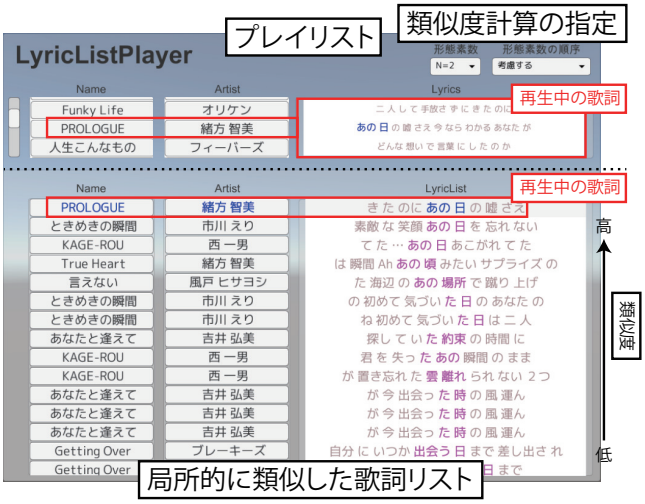

図 1. LyricListPlayer の実行画面

本論文では、そのような歌詞に基づいた音楽再生インタフェース LyricListPlayer を提案する（図 1）。LyricListPlayer の基本機能は、楽曲中の歌声と歌詞の自動対応付けに基づいた、**音楽に連動した歌詞の表示**である。それに加え、歌詞に基づいたシームレスな音楽鑑賞と音楽検索を実現するために、歌詞の**局所的な類似度**に着目し、それを計算して活用する。また、その類似度は、潜在的な意味に基づいて計算することで、文字の表層的な違いを吸収した類似度計算を可能とする。例えば、現在再生されている歌詞が「海」であった場合、その局所的な類似歌詞として、意味的に類似した「夏」「波」「砂」「太陽」等も検索可能とする。

従来、歌詞の自動対応付けに基づく音楽再生インタフェース LyricSynchronizer [5] や、曲・アーティ

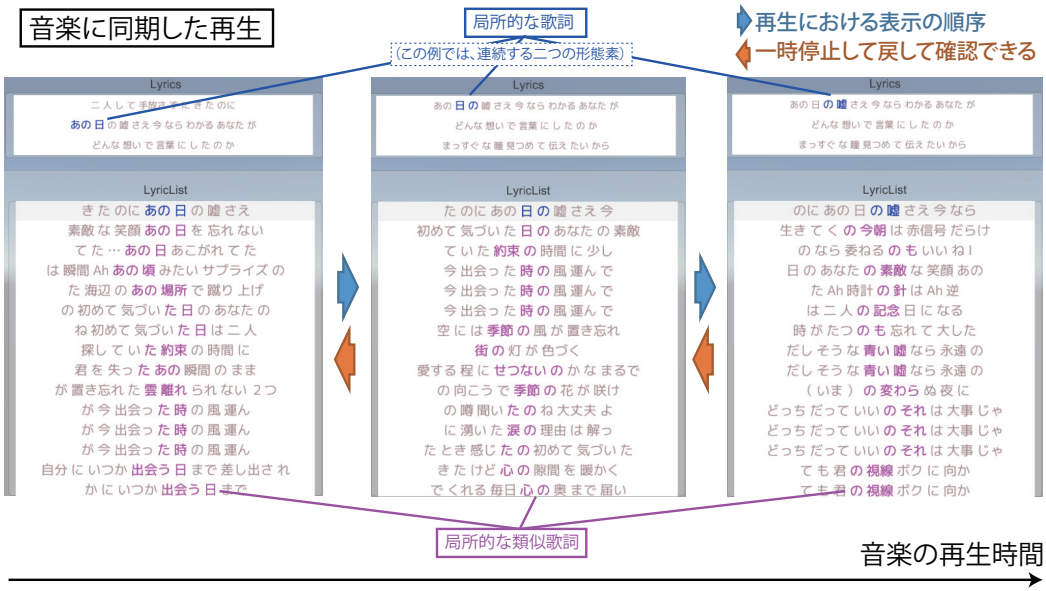

図 2. 歌詞の自動同期結果と局所的な類似歌詞が、音楽の再生が進むにつれて変化する様子

スト単位の歌詞の類似度に基づいた音楽検索インタフェース LyricsRadar [16]、歌詞の特徴的な単語を静的にハイパーリンクさせるシステム Hyperlinking Lyrics [4] が提案されてきたが、リアルタイムで局所的でシームレスな音楽鑑賞・音楽検索は実現されていなかった。LyricListPlayer は、歌詞の自動対応付け機能と、局所的で潜在的な歌詞の類似度を組み合わせることで、シームレスな音楽鑑賞・音楽検索を実現する。具体的には、歌詞に基づいた新しい音楽再生インタラクションとして、局所類似している歌詞を次々に再生する**ザッピング**を可能とする。

2 LyricListPlayer: 歌詞の局所的な類似度に基づいた音楽再生インタフェース

図 1 に LyricListPlayer の実行画面を示す。画面上部に音楽プレイリスト（再生可能な楽曲の一覧）と再生中の歌詞が表示される。そして、画面下部には再生中の局所的な歌詞に対する、潜在的な類似度が高い局所的な歌詞の一覧が表示される。本論文では以降、これを「歌詞リスト」と呼ぶ。一番上が再生中の歌詞であり、それ以降には、再生中の楽曲以外の曲の部分歌詞が、上から類似度順に表示される。

以降、LyricListPlayer の三つの機能を説明する。なお、本論文で用いた図（スクリーンショット）では、全て RWC 研究用音楽データベース（ポピュラー音楽）[20] から 20 曲の楽曲（RWC-MDB-P No.1〜No.20）をプレイリストとして用い、同様にそれらから生成した歌詞リストを表示した結果を示す。

2.1 音楽に同期した歌詞の表示機能（音楽鑑賞）

カラオケのように楽曲の再生と同期して、色が変わる歌詞を表示する機能である。歌詞は、再生されている箇所が着色されて動的に表示され、その箇所に類似した歌詞も同様に着色される（図 2）。

カラオケや LyricSynchronizer [5]、または、Kinetic Typography（歌詞のアニメーション）と異なる点は、色が変わる範囲が限定的である点にある。具体的には、歌詞の形態素を一単位として、再生中の形態素から後続する N 個の連続した部分歌詞（$N = 1$〜5）だけを着色した。

このような形態素単位の可視化により、「今どこを再生しているか」だけでなく「どの範囲で類似歌詞を検索しているか」という情報を両立させて提示できる。歌詞の着色範囲 N は、インタフェース右上のプルダウン（図 1）でいつでも変更可能であり、リアルタイムに変更が反映される。

現在の実装では、画面右上の歌詞をクリックすることで該当の行に相当する時間から再生する（冒頭の場合、曲の先頭から再生）。

2.2 音楽に同期した類似歌詞の表示機能（音楽鑑賞・音楽検索）

音楽に同期した歌詞の表示機能と同様に、再生中の N 個の連続した部分歌詞（$N = 1$〜5）に類似した、他の楽曲の部分歌詞の一覧（歌詞リスト）を表示する。この際、形態素の出現順序を考慮する類似度計算と、出現順序によらない類似度計算の二種類を実装した。出現順序の考慮の有無は、インタフェース右上のプルダウン（図 1）でいつでも変更可能であり、リアルタイムに変更が反映される。

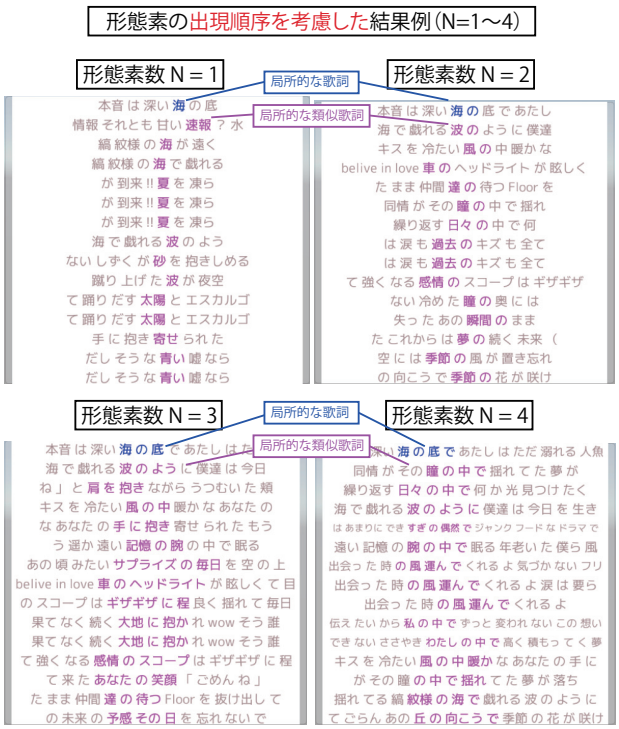

図 3. 類似度計算のための形態素数 N を 1〜4 に変化させた場合（形態素の順序を考慮した類似度計算）

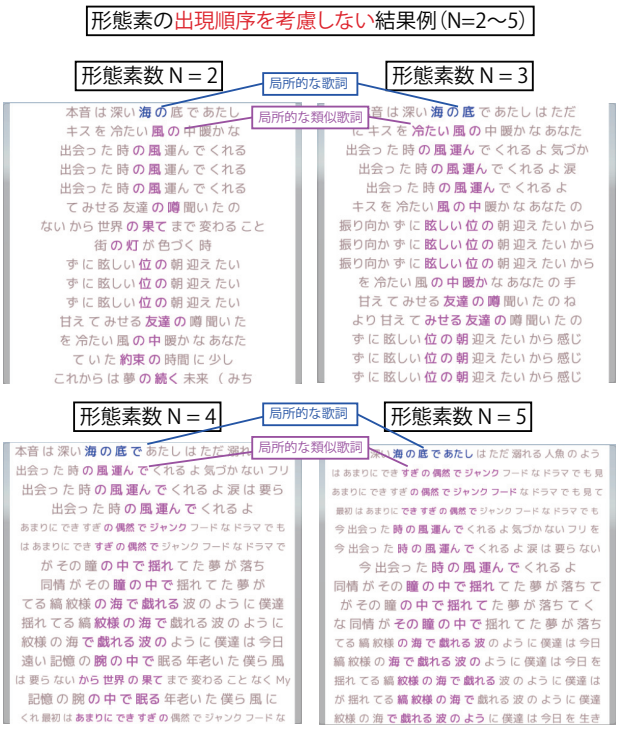

図 4. 類似度計算のための形態素数 N を 2〜5 に変化させた場合（形態素の順序を考慮しない類似度計算）

図 3 に形態素の出現順序を考慮した $N = 1$〜4 の例、図 4 に形態素の出現順序によらない $N = 2$〜5 の例を示す。最上段の青色で着色された形態素は再生中の歌詞、つまり検索クエリに相当する。2 段目以降のマゼンタで着色された形態素が類似歌詞であり、それぞれ、該当箇所前後の歌詞も表示することで、使われた文脈も合わせて可視化した。図に示すように、N を変えると歌詞リストも変化する。

出現順序を考慮する類似度計算では、歌詞の印象の時系列変化が反映されることを意図している。それに対して、出現順序を考慮しない計算は、大まかな印象が反映されることを意図しており、違った観点からの検索結果が得られることを期待した。

ここで、歌詞リストは再生とともに随時変化するため、歌詞リスト全てをリアルタイムに見通すことは難しい。より詳細に確認するために、本論文では、一時停止した上で、着目している歌詞を時間的に前後させる機能を実装した。現在の実装では、キーボードの下矢印で一時停止・再開、左右の矢印で時間を前後させるように実装した。

このような、時々刻々と変化する部分的な類似をどのように可視化すべきかは、議論や今後の検討の余地がある。再生時点の類似歌詞だけでなく、過去や未来も同時に可視化する方法もあるが、意味的な理解に時間がかかると考えたため、一時停止に基づいたインタラクションとして実装した。

2.3 ザッピング再生インタラクション（音楽鑑賞・音楽検索）

音楽に同期して表示される歌詞リストのうち、聴いてみたい類似歌詞があれば、その曲を該当の箇所から再生することができる。現在の実装では、該当の行をクリックすることで動作する。

図 5 にその様子を示す。一つ目の楽曲から、局所的に類似した別の曲の歌詞を再生しながら、さらにそこから類似した歌詞を再生する、といったザッピングが可能である。本研究では、このインタラクションをザッピング再生インタラクションと呼ぶ。

ザッピングは、一時停止して時間を前後させながら、ゆっくりとどの曲に飛ぶかを考えても良いし、再生中に気になった曲をクリックして、リアルタイムに、歌詞を渡り歩くように再生することもできる。

3 LyricListPlayer の実装

LyricListPlayer が入力として想定している歌詞は、漢字かな混じり文のテキストファイルであり、英語や記号を含んでも良い。全ての分析や類似度計算は事前に行い、全ての形態素に対して、部分類似度が高い上位 200 箇所を、それぞれインデックスとして保持している。

このように部分類似度の上位のみを保持しているため、歌詞リストに表示する楽曲を変更する場合に限り、部分類似度の上位を再計算する必要がある。

図 5. 局所類似歌詞をクリックすることで、該当部分から再生できる（ザッピング再生インタラクション）

現在の実装においては、前述した 20 曲に関して、総形態素数が 4133（平均 206.65）であり、全形態素間の類似度算出に要した時間は 260.77 秒 [1] であった（CPU: Intel Core i7-4500U 1.8GHz）。

3.1 形態素解析

入力された歌詞は、形態素解析 MeCab [11] によって形態素列に変換する。類似度計算においては、形態素の原形を用いた。例えば、「逢っ/た」という二つの形態素からなる歌詞の場合、「逢う/た」という二つの原形を計算に用いる。

歌詞のトピック分析に関する先行研究 [16] では、記号などを除き、名詞、動詞、形容詞のみを抽出して用いていた。それに対して本研究では、記号や助詞なども歌詞を特徴付ける場合を考慮し、それらも含めて学習及び類似度計算に用いた。歌詞という言語を処理するにあたって、どういった分析方法を取るのが最適かは、今後の検討課題である。

3.2 音楽と歌詞の自動対応付け

楽曲中の歌声と歌詞テキストの対応付けには、基本的には LyricSynchronizer [5] と同様の手法を用いた。ただし、日本語音素に加え、英語音素も含めて音響モデル（monophone HMM）を新たに学習した。またフィラー検出、歌声区間検出（VAD）、及び摩擦音検出は行わなかった。

日本語の読みは MeCab [11] で読みを推定し、それを音素列に変換した。また、英語の読みは The CMU Pronouncing Dictionary [1] を用いて得た。

3.3 形態素のモデル化

形態素同士の類似度を計算するために、まず歌詞が潜在的に持つ代表的な意味（トピック）を潜在的ディリクレ配分法 (LDA) [2] で求め、各形態素をトピック分布に変換する。トピックモデル構築のための学習データには、日本の音楽チャートであるオリコン [2] で 2000〜2008 年の上位 20 位以内に登場した楽曲 1896 曲の歌詞を用い、形態素及び英単語・記号数（異なり語数）は 18995 であった。

通常、トピック分布は楽曲毎に推定するが、ベイズ推定では、各楽曲におけるトピック分布にはディリクレ分布を仮定し、その事後分布を推定することになる。ディリクレ事後分布のパラメータは、個々の単語（形態素）における各トピックの仮想的な観測回数とみなすことができる。この性質を利用して、事後分布のパラメータを楽曲毎に足し合わせて、楽曲集合のトピック分布を推定する研究がある [24]。

本研究では逆に、個々の形態素における各トピックの負担率を形態素のモデルとして用いる。トピック数は $K = 100$ とし、周辺化 Gibbs サンプラー [7] によって学習した。トピックと単語のディリクレ分布のハイパーパラメータは、それぞれ 1、0.1 とした。

3.4 類似度計算（出現順序を考慮する場合）

連続する形態素 N 個同士の類似度を計算する上で、出現順序を考慮する際、比較する形態素の数が同一であるとし、まず各形態素間の距離を求めた。ここで、形態素 A と形態素 B の距離には、形態素 A のトピック負担率と形態素 B のトピック負担率との分布間距離（対称カルバック・ライブラ距離：KL2）を計算して推定した。

このようにして得られた N 個の距離から、本研究ではそれらの中央値を算出し、その逆数 [3] を類似度とした。中央値を用いることで、形態素間の距離が少ない事例が多いければ、それが重視されるため、平均を取るよりもロバストに計算できるためである。

3.5 類似度計算（出現順序によらない場合）

出現順序を考慮しない場合の類似度計算では、前述したベイズ推定の考え方から、各形態素の負担率を足し合わせて N 個分の形態素のトピック分布として扱う。その後、KL2 を計算して推定した。

この方法では、形態素数の違いを吸収して類似度計算できるため、例えば、歌詞の行同士の類似度を推定することも実装可能である。

[1] 形態素毎に全類似度をファイル出力した時間を含む。また実際には、同一ペアに対する計算は省くことができる。

[2] http://www.oricon.co.jp/

[3] 距離の最小は 0 なので、1 を足してから逆数を取った。

4 LyricListPlayer の運用可能性

歌詞の局所的な類似度でザッピングする音楽鑑賞は、既に知っている少数の曲をプレイリストとして聴く際に、歌詞や楽曲間の関係について、これまでになかった気付きを得る目的で使用できる。また、大量の楽曲を聴く鑑賞スタイルにおいても適合しやすく、特に、あまり知らない（初めて聴く）曲を連続で再生する場合に活用できる。

歌詞に基づいた検索は、音楽ジャンルや歌手の声質が異なっても、同じジャンルの歌詞に出会える可能性がある。さらに、部分的に似ている歌詞を検索できるため、全体の雰囲気は異なるが似ているところもある歌詞の検索にもつなげやすい。

一方、1 曲をじっくり聴く鑑賞スタイルであっても、新たな発見につながる可能性がある。類似歌詞が表示される歌詞リストを鑑賞することで、歌詞の意味的な鑑賞を支援できると考えられる。

5 関連研究

歌詞に基づいた音楽鑑賞、音楽検索そして演奏支援に関する従来研究があった。また、複数の音楽コレクションを部分的に再生するインタフェースの研究があったため、本研究との関連を整理して述べる。

5.1 歌詞に基づいた音楽鑑賞

従来、音楽再生において歌声に同期した歌詞テキストを表示させる研究として、LyricSynchronizer [5] が提案されている。また、合唱生成における再生時刻指定や区間分割に歌詞を活用する Unisoner [17] や、歌詞の同期表示を拡張した Kinetic Typography 生成を支援するインタフェース TextAlive [10] も提案されている。しかしこれらは、複数の楽曲をザッピングする目的で使用することはできなかった。

5.2 歌詞に基づいた音楽検索

歌詞の内容に基づいた検索や推薦については、歌詞に基づいた音楽検索 Web サイト [4] があった。タイトルなどの書誌情報や歌詞テキストの全文検索に加え、年代や音楽ジャンル、ランキングに基づいた検索や音楽聴取者による歌詞コメントに基づいた検索 [5] が可能である。

また、歌詞の類似度に基づいた検索インタフェース LyricsRadar [16] や LDA を用いた歌詞と旋律による楽曲検索 [3] は潜在意味（トピック）に基づいた検索を可能とし、歌詞の特徴的な単語をハイパーリンクさせるシステム Hyperlinking Lyrics は、複数の楽曲の歌詞中に共通して登場するフレーズ（キーワード ）間にハイパーリンクを張ることができる。また、音楽特徴量と歌詞を用いたムードやジャンルの識別に関する研究 [12, 15] があった。LyricListPlayer は、これらの研究と比較して、リアルタイムで局所的な音楽再生が可能であり、検索と鑑賞がシームレスに連動している点で新しい。

5.3 歌詞に基づいた演奏支援

歌を歌ったり弾き語りをすることも、歌詞に意識を向ける上で有効である。Song Prompter [13] は、ベース音と MIDI ドラム音を伴奏として再生しながら、それに同期して、歌詞、コード、ビート位置、小節区切り位置を横スクロール表示することで、ユーザーが曲に合わせて歌唱演奏することを支援する。また、VocaRefiner [14] は歌詞を活用して複数回の歌唱を効率的に録音するためのインタフェースであり、さらにそれら録音の優れた部分を統合して一つの歌声を作る歌声生成インタフェースである。

創作を意識することで、鑑賞体験をより豊かにできる可能性があるため、今後検討の余地がある。

5.4 複数楽曲の部分再生に関する研究

歌詞情報を用いていない関連研究として、複数の楽曲の楽曲間を類似フレーズでなめらかにつないで再生する Song Surfing [23] と、連続する二つの曲が滑らかにつながるようテンポを伸縮させる DJ システム [9] がある。また、大量の楽曲の視聴支援として、Musicream [6] では曲調の類似性に基づいたザッピングを提案していた。PlaylistPlayer [24] では、音響特徴や再生回数などに基づいた楽曲の部分再生を提案していた。

6 おわりに

本論文では、歌詞に基づいて音楽ライブラリをリアルタイムにザッピングできる音楽再生インタフェースを提案した。再生中の部分的な歌詞に類似した、他の曲の部分歌詞がリアルタイムに一覧表示されることで、複数の音楽の歌詞を横断的に閲覧する可能性を持つ。歌詞同士の類似度には、トピックモデルに基づいた潜在的な意味を活用することで、表層的な文字が異なっていても計算を可能にした。図 3 や図 4 を見ると、潜在的な意味が似ている歌詞が、検索可能であると考えられる。

今後は、連続した形態素ではなく歌詞の行単位の類似度を用いたり、歌詞のありがち度 [24] を用いたりする応用が考えられる。また、ボーカルの声質や音楽音響的な特徴を考慮する発展可能性がある。その他、日々更新される大規模音楽データにおける運用能力、ユーザインタフェースの改善、歌詞の言語的特徴に特化した手法の開発などが検討課題である。

[4] Uta-Net (`http://www.uta-net.com/`), J-Lyric (`http://j-lyric.net/`), うたまっぷ (`http://www.utamap.com/indexkasi.html`), JOYSOUND (`https://www.joysound.com/web/search`), 等。

[5] うたまっぷには、「応援」「恋愛」「春」など場面毎に、その場面へのふさわしさに関するランキングがある。

謝辞

本研究の一部は JST CREST「OngaCREST プロジェクト」の支援を受けました。また、RWC 研究用音楽データベースを使用しました。

参考文献

[1] Speech at CMU. The CMU Pronouncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[2] D. M. Blei, *et al.*. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 2003.

[3] E. Brochu and N. de Freitas. "Name That Song!":（略）. In *Proc. of NIPS2002*, 2002.

[4] H. Fujihara, *et al.*. Hyperlinking Lyrics:（略）. In *Proc. ISMIR 2008*, 2008.

[5] H. Fujihara, *et al.*. LyricSynchronizer:（略）. In *IEEE Journal of Selected Topics in Signal Processing*, 2011.

[6] M. Goto and T. Goto. Musicream:（略）. In *Proc. ISMIR 2005*, 2005.

[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *Proc. PNAS*, Vol. 1, 2004.

[8] Y. Hu, *et al.*. Lyric-based Song Emotion Detection with Affective Lexicon. In *Proc. ISMIR 2009*, 2009.

[9] H. Ishizaki, K. Hoashi, and Y. Takishima. Full-automatic DJ mixing with optimal tempo adjustment based on measurement function of user discomfort. In *Proc. ISMIR 2009*, 2009.

[10] J. Kato, *et al.*. TextAlive:（略）. In *Proc. ACM CHI 2015*, 2015.

[11] T. Kudo. MeCab:（略）. http://mecab.sourceforge.net/.

[12] C. Laurier, *et al.*. Multimodal Music Mood Classification using Audio and Lyrics. In *Proc. ICMLA*, 2008.

[13] M. Mauch, *et al.*. Song Prompter:（略）. 情報処理学会 インタラクション 2011 論文集, 2011.

[14] T. Nakano and M. Goto. VocaRefiner:（略）. In *Proc. SMAC-SMC2013*, 2013.

[15] R. Neumayer and A. Rauber. Integration of Text and Audio Features for Genre Classification in Music Information Retrieval. In *Proc. ECIR*, 2008.

[16] S. Sasaki, *et al.*. LyricsRadar:（略）. In *Proc. ISMIR 2014*, 2014.

[17] K. Tsuzuki, *et al.*. Unisoner:（略）. In *Proc. ICMC-SMC 2014*, 2014.

[18] K. Watanabe, *et al.*. Modeling Structural Topic Transitions for Automatic Lyrics Generation. In *Proc. PACLIC 2014*, 2014.

[19] 星野 悦子. 歌の聴取印象と再認記憶: 言葉とメロディの関係を探る. 情報処理学会研究報告 音楽情報科学研究会, 第 2002 巻 of *2002-MUS-45*, 2002.

[20] 後藤 真孝 他. RWC 研究用音楽データベース:（略）. 情報処理学会論文誌, 45(3), 2004.

[21] 森 数馬. 日常の音楽聴取における歌詞の役割についての研究. 対人社会心理学研究, 10, 2010.

[22] 森 数馬 他. 演奏音の印象に及ぼす歌詞の影響について – 印象評定値の個人差に基づく検討 –. 日本心理学会第 72 回大会, 2008.

[23] 堀内 直明 他. Song Surfing:（略）. *PIONEER R&D*, 17(20), 2007.

[24] 中野 倫靖 他. 確率的生成モデルに基づく音楽の類似度とありがち度の推定に関する検討. 情報処理学会 研究報告 音楽情報科学研究会, 2014-MUS-104, 2014.

未来ビジョン

音楽のデジタル化により、アクセス可能な楽曲が単調増加していく現在では、音楽聴取者が好みの楽曲に出会うこと、音楽制作者が自分の曲を届けることを支援することは、音楽産業や文化の観点から重要であると考える。そのような問題を解決する一つの方法が、音楽推薦や音楽検索に関する技術やインタフェースであり、これまで数多く研究されてきた。

しかし、仮に「好みかもしれない」楽曲を推薦されたり、検索できても、実際に好みの楽曲であるかどうかを判断するには、聴くしかない。したがって、限られた時間の中で「どう聴くか」「いつ聴くか」は、これからの音楽鑑賞・検索における重要な課題である。すなわち、本論文の 5.4 節や文献 [24] で議論されているように、部分再生、高速鑑賞、情報要約、ダイジェスト再生、隙間時間活用等の、聴き方に関する研究が重要となる。

本論文は、その中で「どう聴くか」に焦点を当てたものであり、歌詞を対象とした。歌詞は、音を聴かなくても内容を理解でき、その理解に音楽的な能力を必要としない点で、一覧表示して理解可能な要素である。したがって、音楽的な能力を持たないユーザを対象にでき、大量の音楽を扱う上で応用可能性が高い。

また、本インタフェースの特徴の一つは、部分的な類似度を扱い、その着目する範囲（形態素数）をインタラクティブに変更できる点にある。楽曲間類似度や楽曲集合の類似度、また固定的な部分類似度は扱われたことがあったが、「どれくらいの部分を使うか」を決定できるインタラクションはこれまでに研究がなかった。部分類似度の計算は、時系列メディアにおける嗜好や印象等を特定時刻に着目して扱う上で、今後の重要性が高まってくると考える。

以上、未来の音楽鑑賞インタラクションには「どう聴くか」と「部分的類似度」が重要になると考え、LyricListPlayer を実装・提案した。