

## スペル修正技術を用いた楽曲検索システム

鈴木 督史<sup>†1</sup> 川越 恭二<sup>†2</sup>

本稿では、スペル修正技術を用いた楽曲検索システムについて提案する。現在、利用者が問合せとしてハミングを入力する、楽曲検索システムが存在している。しかしながら、利用者がいつも正確にハミングをシステムに対して入力することが可能とは限らない問題が考えられる。本稿では、この問題に対して文書検索におけるスペル修正の技術をもちこむことによって解決を図る。はじめに、本提案システムは、旋律情報を分割する。次に、分割した旋律情報に対して、文書における“単語”と“文字”のモデルを当てはめる。最後に利用者の入力した問合せに対してスペル修正の技術を用いる。

### A Music Retrieval System with Spelling Correction Technique

MASAFUMI SUZUKI<sup>†1</sup> and KYOJI KAWAGOE<sup>†2</sup>

In this paper, we describe a novel music information retrieval system using spelling correction technique. The current music retrieval systems with humming enable a user give a query with humming. However, there is a problem that a user cannot always input a complete melody to the system with humming. In order to solve the problem, we employ the spelling correction technique, which is used usually in a document retrieval system. Our system first converts user's query by humming into melody information then in order to apply the spell correction, “term” and “characters” are extracted from the melody. Finally, our system can correct the mistake of the user's query by the spelling correction technique.

<sup>†1</sup> 立命館大学理工学研究科

Graduate School of Science and Engineering, Ritsumeikan University

<sup>†2</sup> 立命館大学情報理工学部

College of Information Science and Technology, Ritsumeikan University

### 1. はじめに

Webの普及によって情報検索システムは我々の身近なものとなった。特に文書検索システムは検索エンジンに代表されるように、広く使われている規模にある。既存の文書検索システムは、利用者は目的の文書に関連する文字列であるキーワードを検索システムに対して入力する。キーワードによって指定された問合せに対して、システムは利用者の目的の文書と推測される文書の一覧を表示する。

このように、検索システムを利用する場合、利用者がシステムに何らかの情報を入力する機会が多く存在している。しかし、利用者が入力すべき情報を誤ってしまう場合が考えられる。利用者が情報の入力を誤った場合、検索システムは利用者が望む文書検索をすることが不可能となる。このような問題に対して、優れた検索システムでは様々な方法を用いて解決が為されている。その一例として、利用者の入力の誤りを訂正するアプローチが存在する。もしGoogleの検索窓に“speling”と入力したならば、即座に“もしかして: spelling”と表示される。この例のような、単純なスペル修正のアプローチは、解決手段として、最も基本的なもののひとつであると考えられ、非常に有効に作用している。優れたシステムは、利用者のミスに対して寛容である。つまり、優れた検索システムを設計するためには、利用者のタイプミスやうろ覚えといった、よくある要因によって引き起こされるミスに対して、何らかの方法を用いて対策する必要がある。

一方、マルチメディア情報検索においては内容検索が注目されている。内容検索では、利用者の目的とする情報の形式と同じ形式の情報を問合せとして利用する。例えば、画像検索では写真や絵画の情報を、楽曲検索では楽曲データを問合せとして利用する。なぜならば、マルチメディア情報検索では、問合せを文字列で表現することが困難である場合があるからである。楽曲検索においても内容検索を用いる情報検索システムが試みられている。例えば、利用者がハミングを用いて検索するシステムが多数存在<sup>2)8)5)6)</sup>する。楽曲検索でも内容検索を用いるアプローチが広く試みられている。しかし、内容検索において、冒頭のWeb検索の例のような、利用者が問合せの入力を誤った場合についての解決は未だになされていない。

そこで、本稿ではハミングを用いた楽曲検索システムにおける、問合せ修正方法を提案する。本稿では問合せの修正の手法として、文書検索におけるスペル修正の技術を利用する。スペル修正とは古典的な入力修正の手法であり、文書処理システムなどで運用されている。提案する楽曲検索システムでは、旋律情報に対してスペル修正の技術を用いる。そのため、

旋律情報に文書のモデルを適用する。システムは旋律情報を分解し、それらを文書における単語と文字にモデルに当てはめる。本稿では、システムを実装し、得られた知見について報告する。

## 2. スペル修正技術を用いた楽曲検索システム

ハミングを用いた楽曲検索について、既存で実用化された方法が存在している<sup>2)8)5)6)</sup>。KOSUGI ら<sup>2)</sup>が提案されている手法を用いる場合、90%程度という高い検索精度が実現可能である。しかし、1章で述べた通り、利用者が問合せの入力を誤った場合、検索結果の出力は困難となる。本稿では、利用者が問合せの入力を誤った場合においても、問合せを修正するアプローチを用いて、検索精度を高める。そのため、本稿では、検索システム本体は、KOSUGI ら<sup>2)</sup>が用いられている方式を利用する。

本稿ではスペル修正技術を楽曲に適用する方法について提案を行う。スペル修正技術は古典的な技術であるが、旋律情報に対するスペル修正技術は未だ試みられていない。本提案システムは、旋律情報に対して文書のモデルを与えることによって、スペル修正を旋律情報に適用することを可能としている。

旋律情報に対する、文書モデルの適用は、KOSUGI ら<sup>2)</sup>で検索のためにおこなわれる旋律分割方法に対して、更なる分割処理を加えるによって可能となる。旋律情報の分割による旋律情報に対する文書モデルの適用手法について 2.2 で詳しく述べる。次に文書モデルを与えた場合においても、旋律情報の比較については特徴ベクトル間の距離を用いた比較が必要となる。特徴ベクトル間距離の算出方法は、既存手法を用いる。本手法における特徴ベクトルを用いる必要性および、特徴ベクトル間距離の算出方法は 2.3 節で詳しく述べる。最後に、文書モデルを与えた旋律情報に対するスペルチェック適用方法について、2.4 節で詳しく述べる。

### 2.1 概要

スペル修正技術を用いた楽曲検索システム概要を図 1 に示す。本システムを実行するために、あらかじめ楽曲 DB を構築しておく。楽曲 DB は楽曲の旋律情報を検索およびスペル修正に有用な形式に変換した情報を格納する。構築した楽曲 DB に対しては、楽曲の旋律情報を登録しておく。これは、一般的に楽曲をハミングで検索する場合、楽曲の旋律情報が使われるからである。楽曲の旋律が入力されると、システムは旋律情報に文書モデルを当てはめる。このために旋律情報を分割する。次にシステムは、処理の高速化のために、旋律情報の特徴ベクトルをもとに索引付けを行う。特徴ベクトルおよびモデル化の手順は 2.2

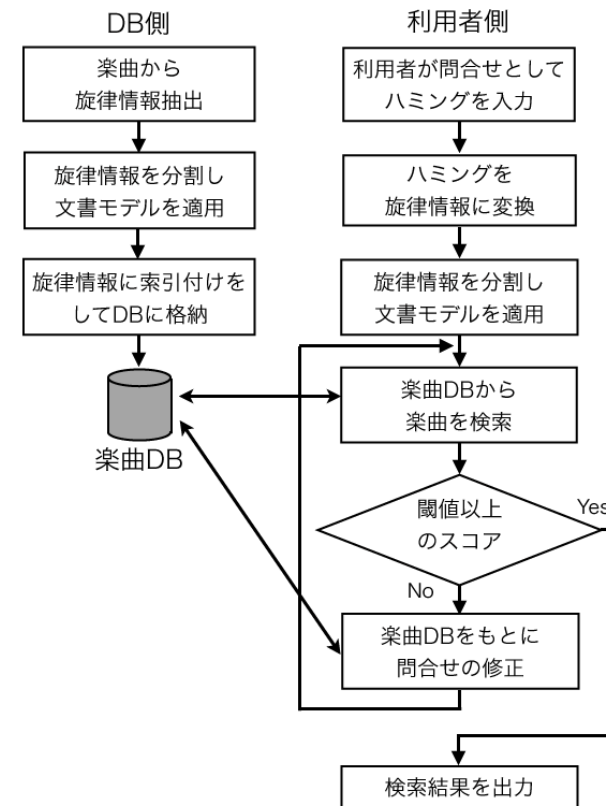


図 1 スペル修正技術を用いた楽曲検索システム概要  
Fig.1 Abstract of A Music Retrieval System with Spelling Correction Technique

節で、索引付けの手順を 2.4.2 節で詳しく述べる。楽曲 DB を構築することによって、システムは利用可能となる。

利用者が本システムを利用するとき、利用者はまずシステムに対してハミングを問合せとして入力する。システムがハミングを受け取ると、ハミングの情報を旋律情報に変換する。次に、システムは楽曲 DB と同様の手順で変換した旋律情報に文書モデルを適用する。次にシステムは、楽曲 DB をから楽曲の検索を行う。楽曲の検索は 2.3 節で定義する特徴ベク

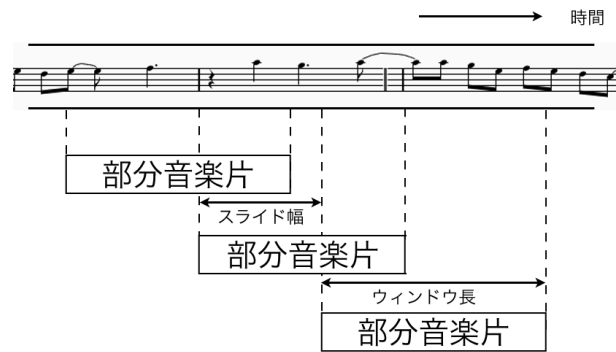


図2 スライディングウィンドウ方式  
Fig.2 Sliding Window Technique

トルの距離を用いて、その距離をスコアとして用いる。この時点で、定めた閾値よりもスコアが高ければ、システムは検索結果を出力する。逆に、スコアが低く適切な検索結果が出力されないとと思われる場合、システムはスペル修正の技術を用いて利用者の問合せの修正を試みる。スペル修正の手順は、2.4節で詳しく述べる。

## 2.2 文書モデルの適用

スペル修正の技術を旋律情報に対して適用するために、本システムは旋律情報を文書のモデルに当てはめる。文書は通常、複数の“単語”によって構成されている。また、“単語”は複数の“文字”によって構成されている。旋律情報を文書のモデルに当てはめる場合、旋律情報をこれらの“単語”と“文字”の単位に分割する必要がある。旋律情報の分割方法に、楽曲検索や類似楽曲探索において、既にいくつか存在している<sup>1)2)</sup>。本システムでは、“単語”の単位の分割に、楽曲検索で一般的に用いられる、スライディングウィンドウ方式を利用する<sup>7)</sup>。スライディングウィンドウ方式について図2に示す。本システムでは、分割されたウィンドウを部分音楽片と呼ぶ。スライド幅をウィンドウ長より短くすることによって、部分音楽片は互いに重なり合う冗長な情報となる。このことによって、利用者は楽曲の任意の部分のハミングを問合せとして入力した場合でも、楽曲の検索が可能となると考えられる。

次に、抽出された部分音楽片を更に分割し、文字の単位に分割する。既存の類似楽曲検索において、ウィンドウを更に分割する場合においては、一般的に音符の出現を基準としたN-gramが用いられる<sup>1)10)</sup>。しかし、ハミングを用いた楽曲検索において、音符の出現を基

準としたN-gramを用いる場合、正しい出力は困難となる。なぜならば、よく知られているように、人の声をもとにしたピッチ検出に関しては、未だに決定的な方法が存在しておらず、様々なノイズが混入されてしまうからである。そのため、余分な旋律情報が混入したり、逆に必要な旋律情報を取りこぼしてしまったりする問題が発生する。そのため、音符の出現を基準としたN-gramを用いる場合、分割の単位として、部分で分割のタイミングが、ハミングをもとにした旋律情報と、手動で入力した旋律情報で大きく異なってしまふ。例えば、人間の入力したデータの音階の揺らぎが誤って旋律情報として入力された場合、余分な旋律情報が数多く入力されてしまふ。このデータに対して音符の出現をもとにしたN-gramでの分割を行った場合、例え直感的によく似ている旋律であったとしても、正しく比較を行うことが困難となる。そのため、本システムでは、固定時間軸での旋律情報の分割を行い、その方法で分割したそれぞれの要素を文書の“単語”における“文字”として扱う。実際の文字情報の比較は2.3節で詳しく述べる。

## 2.3 特徴ベクトル付与

2.2章で得られた文書モデルを適用した部分音楽片に対してそのまま、スペル修正を適用する場合、更なる問題が発生する。旋律情報は、文書と比較して曖昧であり、また利用者の入力したハミングには音階の揺らぎが存在する。そのため、旋律情報を比較する場合完全一致する場合は稀である。この問題を解決するために、部分音楽片に対して以下に示す特徴ベクトルを与え、特徴ベクトル間の距離を用いることによって類似比較を行う。

本システムでは、旋律情報に対して特徴ベクトルを与える方法として既存の手法を用いる。特徴ベクトルとして、音高の時系列を表現する“音高推移特徴ベクトル”<sup>2)</sup>を用いる。音高推移特徴ベクトルにおける、音高は、部分音楽片の音高の中心点を基準に相対的な音高を用いる。

### 2.3.1 特徴ベクトル間比較

既存手法と同様に、音高推移部分特徴ベクトル<sup>2)</sup>を用いて比較を行う。スライディングウィンドウ方式を用いた場合でも、ウィンドウの先頭と利用者の問合せの先頭が非一致となる場合が考えられる。この問題を防ぐため音高推移部分特徴ベクトルを用いる。

音高推移部分特徴ベクトル間距離の算出は、まず、比較する特徴ベクトル双方の次元を正規化する。このときの次元数は、次元の少ない方のベクトルの次元数とする。このことによって、入力されたハミングの揺らぎが吸収されると同時に、入力されたハミングの揺らぎが大きい場合での計算速度の向上が期待が可能である。これら双方の特徴ベクトルのマンハッタン距離を次元数で割ったものを距離とする。

スコア付けは、部分ハミング片間 OR 検索<sup>2)</sup>を用いる。ハミング  $h$  から生成した部分音楽片数を  $m$ 、比較する楽曲  $a$  の部分音楽片数を  $n$  とする。比較する楽曲  $a$  の  $i$  番目の部分音楽片と、 $j$  番目のハミングから生成した部分音楽片との、距離を  $d(a_i, h_j)$  としたとき、比較する楽曲  $a$  とハミング  $h$  の距離  $D(a, h)$  を (2.3.1) 式のように表す。

$$D(a, h) = \min_{1 \leq i \leq n, 1 \leq j \leq m} \{d(a_i, h_j)\} \quad (1)$$

ベクトル間距離  $D(a, h)$  が小さければ小さいほど、高いスコアをつける。

## 2.4 スペル修正

現在様々な方式のスペル修正技術を用いたソフトウェア及びライブラリが存在している。有名なものに Hunspell<sup>\*1</sup>や、GNU Aspell<sup>\*2</sup>が存在している。これらの技術は、Mozilla Firefox<sup>\*3</sup> や Opera<sup>\*4</sup> といった Web ブラウザや、OpenOffice.org<sup>\*5</sup> といった、オフィススイートソフトウェアで利用されている。我々はこのような手法をハミングを用いた楽曲検索において適用する。

スペル修正の基本戦略として、レーベンシュタイン距離を用いた手法がよく知られている<sup>3)</sup> <sup>\*6</sup>。レーベンシュタイン距離とは、二つの文字列を比較する時、一方の文字列をもう一方の文字列に変更するための必要最小手順を定量化した距離である。例えば、“speling” から “spelling” の距離は、spel の後に “l” を 1 文字追加すればよいので、この二つの文字列の編集距離は 1 となる。このように、レーベンシュタイン距離は、一方の文字列に挿入、削除置換操作を 1 度行うごとに、1 追加される。このレーベンシュタイン距離を単純に用いた場合、レーベンシュタイン距離が同じ候補が複数存在する場合、どれを優先すべきかわからないという問題が発生する。そのため、文書検索における単語の文書頻度を表す DF 値<sup>4)</sup> を利用する。DF 値は、対象の単語がどの程度一般的であるかの指標となる。このため、編集距離が同様の場合は、DF 値を用いて優先度を与える。

この基本的なスペル修正の正しさについては、確率論で説明することが可能である。利用者の入力した過った入力  $Y$  が与えられた場合、修正可能な確率的に最大である  $W$  を表す場合、(2) 式のように表すことが可能である。

$$\operatorname{argmax}_W P(W|Y) \quad (2)$$

\*1 <http://hunspell.sourceforge.net/>

\*2 <http://aspell.net/>

\*3 <http://mozilla.jp/firefox/>

\*4 <http://www.opera.com/>

\*5 <http://www.openoffice.org/>

\*6 <http://nlp.stanford.edu/IR-book/html/htmledition/spelling-correction-1.html>

この (2) 式はベイズ定理によって、(3) 式と見なすことができる。

$$\operatorname{argmax}_W P(Y|W)P(W) \quad (3)$$

このとき、 $P(Y|W)$  がレーベンシュタイン距離、 $P(W)$  が DF 値に相当すると考えられる。

### 2.4.1 旋律情報におけるスペル修正

本システムでは、スペル修正におけるレーベンシュタイン距離の算出方法として最も伝統的な手法である動的計画法をもとに、その手法を応用した手法を利用する。

編集距離  $LD$  は、二つの文字列のそれぞれ  $i$  番目と  $j$  番目の編集距離  $LD(i, j)$  を求める場合、一つ前の文字までの編集距離  $LD(i-1, j-1)$ 、 $LD(i-1, j)$ 、 $LD(i, j-1)$  を利用することによって算出が可能である。このような、最適解を求める問題の特性は部分問題最適性と呼ばれている。部分問題最適性を持つ時に動的計画法は有効である。

旋律情報に対して、動的計画法を用いる場合 2.3 節で挙げた問題と同様に、旋律情報は曖昧なため、完全一致することは稀である問題が発生する。そのため、文字モデル間の比較にベクトル距離を用いる。文字モデルに与える特徴ベクトル及び比較方法は、比較するベクトルそれぞれを次元の大きい方に合わせて次元正規化を行った後、マンハッタン距離を算出し次元数で割る。そのときの距離が閾値以下の場合、一致すると定める。

### 2.4.2 特徴ベクトルを用いた索引付け

動的計画法は、その計算量が  $O(mn)$  のため、検索対象が数万件となる場合、すべてとの比較を行うのは時間的に現実ではない。そのため、部分音楽片の特徴ベクトルを用いて索引付けを行う。得られた索引を用いて、あらかじめ比較対象を絞り込むことによって、現実的な実行速度を実現する。

索引付けについていくつかの方法が考えられるが、本提案システムでは、N-gram 索引<sup>9)</sup> を利用する。部分音楽片に含まれる文字モデルを N-gram で切り出しデータベースに索引として格納する。スペル修正を行う際に、この索引を用いて比較対象の絞り込みを行う。この時の文字比較は 2.4.1 節と同様にベクトル間距離を利用し、閾値を用いて一致判定を行う。

## 3. 実 験

### 3.1 プロトタイプ

旋律情報に対するスペル修正の技術の有用性を確認するために、プロトタイプを実装した。プロトタイプシステムの概要を図 3 に示す。プロトタイプは、利用者がハミングを旋律情報に変換して記録した MIDI データを入力する。プロトタイプは、ハミングから旋律情報に変換する機能を有していない。このため、Voice to Midi ソフトウェアと呼ばれるソ

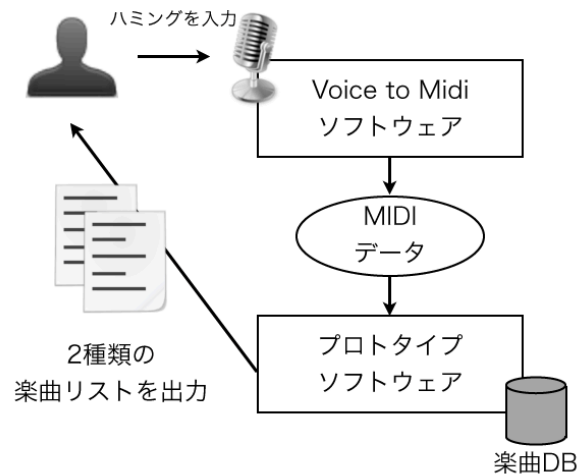


図3 プロトタイプ概要  
Fig.3 Abstract of Prototype

ソフトウェアを用いて MIDI データを用意する必要がある。楽曲 DB に含まれている楽曲を、入力されたデータと距離の近い順番でリストに出力する。この際、利用者はスペル修正を利用するかどうかの選択が可能である。

### 3.2 実験条件

プロトタイプシステムでは、ハミングを旋律情報に変換する機能を既存のソフトウェアである、ソング頼太<sup>\*1</sup>を利用して開発した。ソング頼太は、鼻歌を用いた作曲ソフトウェアである。本実験ではソング頼太に含まれる、鼻歌の鉄人機能を用いて、プロトタイプに入力する MIDI データを作成した。

また、実験に用いる楽曲 DB は、実験目的としてプロの作曲家が制作した旋律情報を利用した、15 から 30 秒の単音の旋律情報 30 曲を利用した。楽曲 DB 作成におけるスライド幅は 2 小節とし、ウィンドウの分割の単位は 2 拍とした。

### 3.3 実験方法と結果

以下に挙げる 2 点の問合せをシステムに入力した場合のスペル修正を用いない場合と用

いる場合の出力を予備実験として比較した。本実験で問合せとして利用する楽曲は、楽曲すべてをハミングとして入力した。

**問合せ A** 正しく検索されると考えられる問合せ

**問合せ B** 一部を意図的に間違えた問合せ

各々の問合せについて、スペル修正を用いる場合と用いない場合で比較を行った。

この予備実験の結果、すべての条件で目的の楽曲がリストの最上位に出力された。検索対象の楽曲が少ないという状況であるため、問合せの一部を意図的に間違えたとしても、目的の楽曲が最上位になったと考えられる。また、**問合せ B** の条件のそれぞれ異なる問合せを用いて、20 回の問合せを試みたところ、そのうち 19 回においてスペル修正によって正しく問合せの修正が行われ、検索結果の出力が可能であった。そのことによって、スペル修正技術が旋律情報に対して適用可能であることを確認した。

## 4. おわりに

本稿ではスペル修正技術を用いた楽曲検索システムについて提案した。システムのプロトタイプを作成することによって、旋律情報に対し、文書のモデルを適用しスペル修正が可能であることを確認した。一方、楽曲検索システムにおいて、スペル修正が有用性については未解となった。これは、本稿での実験で用いる楽曲数が少なすぎたことに起因する。そのため、今後課題として、楽曲データ数を増やした実験を行うことを予定している。実験結果によって、文字単位での分割方法の最適化や、高速化の手法についての提案が可能となると考えられる。

また、旋律情報への文書モデルの適用は既存の文書検索における、多くの知見を利用可能となるため、様々な応用が考えられる。本稿で提案した、旋律情報に対するスペル修正の技術は、検索における問合せ修正の他に、Voice to MIDI や、旋律情報打ち込み支援といった分野に対しても活用可能であると考えられる。

## 謝辞

本稿における実験で利用した楽曲データは、有限会社ト・ヘン 佐野芳彦先生に研究用に作曲していただきました。ここに記して謝意を表します。

\*1 <http://www.dynasys.co.jp/FreeSoft/Srt/index.htm>

## 参 考 文 献

- 1) Cui, B., Jagadish, H.V., Ooi, B.C. and Tan, K.-L.: Compacting music signatures for efficient music retrieval, *EDBT '08: Proceedings of the 11th international conference on Extending database technology*, New York, NY, USA, ACM, pp.229–240 (2008).
- 2) KOSUGI, N.: SoundCompass : A Practical Query-by-Humming System-Normalization of Scalable and Shiftable Time-Series Data and Effective Subsequence Generation, *Proc. ACM SIGMOD International Conference on Management of Data, 2004*, pp.881–886 (2004).
- 3) Manning, C.D., Raghavan, P. and Schütze, H.: *Introduction to Information Retrieval*, Cambridge University Press (2008).
- 4) Salton, G.(ed.): *Automatic Text Processing*, Addison-Wesley Longman Publishing Co., Inc. (1985).
- 5) 市川拓人, 鈴木基之, 伊藤彰則, 牧野正三: 音程特徴量の確率分布を考慮したハミング入力楽曲検索システム (検索・推薦), 情報処理学会研究報告. [音楽情報科学], Vol.2007, No.81, pp.33–38 (20070801).
- 6) 市川拓人, 鈴木基之, 伊藤彰則, 牧野正三: 複数の音程特徴量によるハミング入力楽曲検索システムの高精度化 (スペシャルセッション・歌情報処理 1), 情報処理学会研究報告. SLP, 音声言語情報処理, Vol.2008, No.12, pp.7–12 (20080208).
- 7) 小杉尚子, 西原祐一, 紺谷精一, 山室雅司, 串間和彦: ハミングを用いた音楽検索システム, 電子情報通信学会技術研究報告. DE, データ工学, Vol.99, No.201, pp.49–54 (19990721).
- 8) 西村拓一, 滝田順子, 後藤真孝, 岡隆一: 類似メロディー区間検出による音楽時系列検索の高速化, 情報処理学会研究報告. [音楽情報科学], Vol.2001, No.16, pp.63–70 (20010222).
- 9) 北研二: 確率的言語モデル, 東京大学出版会 (1999).
- 10) 澤井賢一, 平田祥人, 富岡亮太, 合原一幸: 時系列解析手法を用いた楽曲間類似度 (音楽構造解析), 情報処理学会研究報告. [音楽情報科学], Vol.2008, No.78, pp.63–68 (20080730).