クラシック音楽の楽曲と解説文の自動対応付け

服部 友理 † 上垣外 英剛 ‡ 高村 大也 ‡§ 奥村 学 ‡
† 東京工業大学工学院 ‡ 東京工業大学科学技術創成研究院 § 産業技術総合研究所
{hattori@lr., kamigaito@lr., takamura@, oku@}pi.titech.ac.jp

1 はじめに

クラシック音楽の楽曲の解説文には，音楽用語が多用されるため，音楽知識のない読み手にとっては内容の理解が難しいという問題が存在する．クラシック楽曲の解説文は，楽曲の進行に沿い，順序立って主題やコーダなどの楽曲を構成するフレーズに対する説明を行っている．この特徴に着目し，楽曲を聴きながら，再生箇所のフレーズに該当する説明を順次表示することにより，楽曲を通して解説文の理解を支援できると考えられる．一方で，このようなシステムを実現するためには，解説文と楽曲の関連する箇所を対応付けなければならない．本研究では，この課題を解決するために，クラシック楽曲を構成する音楽フレーズと，その楽曲の解説文中のテキストフレーズを自動で対応づけるシステムを提案する．

音とテキストの関連付けに関する研究として，Vijayakumar ら [1] は，音声とそれに関するテキストのペアデータセットから，擬音語に基づく単語分散表現を獲得した．また，高橋ら [2] は，楽曲の歌詞やレビューなどのテキストに含まれる語彙と音響特徴量を線形変換によって関連付け，Web ページの BGM 楽曲を自動で検索するシステムを提案した．本研究は，テキストと楽曲に対してフレーズ単位での対応付けを行うという点でこれらの研究とは異なる．

提案システムは，入力として与えられた楽曲毎の楽譜を分割して得られる音楽フレーズ系列と，解説文を分割して得られるテキストフレーズ系列の対応付けを自動で行う．具体的な手法としては，分類器とマッチングアルゴリズムの 2 つを用いる．分類器は，音楽フレーズとテキストフレーズのペアを入力とし，ペアであるかどうかを二値分類する．この分類器から音楽フレーズとテキストフレーズの類似度スコアを獲得し，マッチングアルゴリズムによるフレーズ系列間の対応付けに利用する．分類器には Random Forest，マッチングアルゴリズムには，ビタビアルゴリズムおよび貪欲法の 2 つの手法を用いた．図 1 に提案システムによる楽曲と解説文の対応付けの例を示す．

また，ベートーヴェンのピアノソナタ楽曲の電子楽譜とその解説文を収集し，101 曲分の音楽フレーズとテキストフレーズを人手で対応付けたペアデータを作成した．このデータを用いて，提案システムによる楽曲と解説文の対応付けの性能評価を行った結果，対応行列の対角線上の対応を正解と仮定したベースラインと比較して，高い F 値が得られた．

2 提案システム

提案システムの概要を図 2 に示す．システムの入力は各楽曲毎の電子楽譜と解説文のペアで与えられ，それぞれを分割して，音楽フレーズとテキストフレーズの時系列データを作成する．各フレーズのベクトル表現を用いて，音楽フレーズ系列とテキストフレーズ系列を対応付けた結果を出力する．ただし，2 つ以上のフレーズ同士が対応する場合があるため，出力は音楽フレーズとテキストフレーズの多対多の対応で構成される．

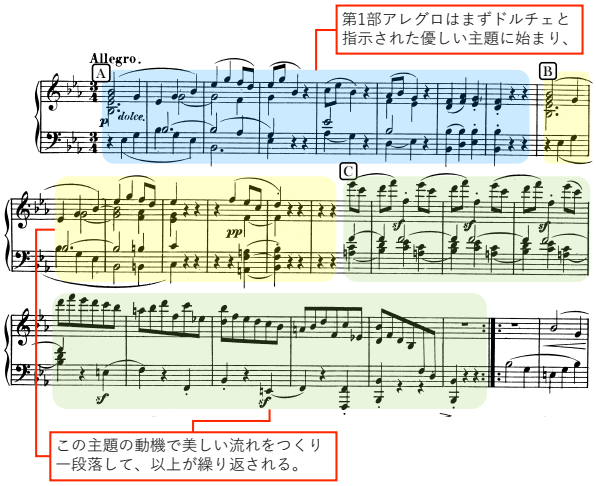

図 1: 提案システムによる楽曲と解説文の対応付けの例 (楽譜上のアルファベットは音楽フレーズ区切り，赤線で囲われた文章は各音楽フレーズに対応するテキストフレーズ．実際には音声データと対応付ける．文章出典：ベートーヴェン (作曲家別名曲解説ライブラリー)[3]，p.358)

2.1 フレーズの作成

本節では，音楽フレーズとテキストフレーズの作成手順を説明する．

2.1.1 音楽フレーズの作成

音楽フレーズは musicXML 形式[1]の電子楽譜から作成する．まず，楽譜作成ソフト MuseScore[2]を用いて，各楽曲の電子楽譜に対して，人手でメロディの自然な区切り位置にアノテーションを付与し，フレーズ区切り付き電子楽譜を作成する (例．図 1 の楽譜)．これらのアノテーション付き電子楽譜から得られる各フレーズの区切り時刻に基づいて，WAV 形式の音声データを各時刻で分割したものを音楽フレーズとする．

[1]https://www.musicxml.com/
[2]https://musescore.org/ja

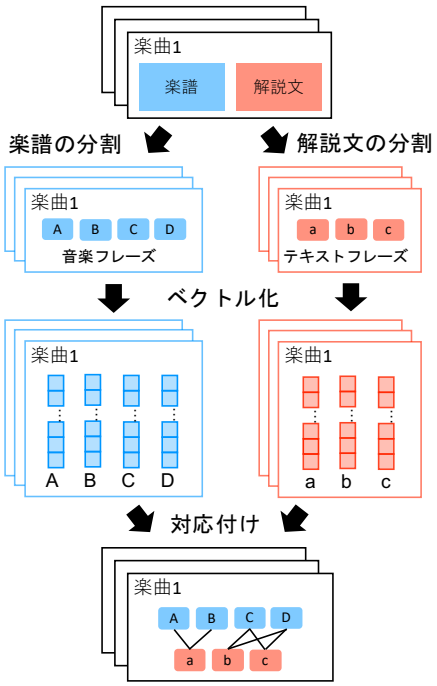

図 2: 楽譜と解説文の対応付けシステム概要

2.1.2 テキストフレーズの作成

解説文の分割例を図 3 に示す．テキストフレーズは，解説文を各楽曲毎にルールに基づいて機械的に分割して作成する．まず，句読点で分割されたフレーズを作り，各フレーズの中に楽曲構成用語が含まれる場合は，その係り先でさらに分割する．楽曲構成用語は「第 1 主題」や「経過部」などの楽曲中を構成するフレーズを指す用語で，あらかじめリストを用意する．係り先は，日本語係り受け解析器 CaboCha[3]で構文解析を行った結果を用いる．

2.2 フレーズのベクトル化

次に，獲得した音楽フレーズとテキストフレーズに対してそれぞれのベクトル表現を計算する．

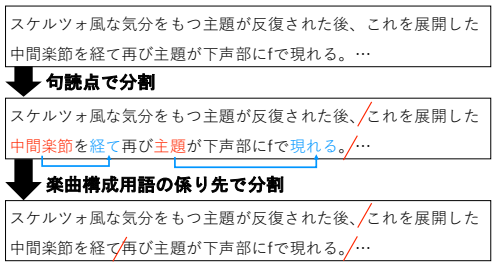

図 3: 解説文の分割例 (赤線はフレーズの区切り位置，赤字は楽曲構成用語，青字は係り先の動詞，青矢印は係り受け関係．文章出典：ベートーヴェン (作曲家別名曲解説ライブラリー)[3]，p.349)

[3]https://taku910.github.io/cabocha/

2.2.1 音楽フレーズのベクトル化

音楽フレーズは，音響特徴量とフレーズ長，位置情報を用いてベクトル化する．音響特徴量には，音楽情報処理の分野において，音色，和音，オンセットの特徴量として用いられている，MFCC，Chromagram，Tempogram の 3 つを使用する [4]．これらの素性を結合して，音楽フレーズのベクトル化を行う．

音響特徴量を用いたフレームベクトルの獲得

音響特徴量を用いた素性として各フレームのベクトルを平均したものを用いる．フレームは，音響信号から得られた時系列の音響特徴量を短く分割した単位である．以下の手順で word2vec[5] を学習し，フレームのベクトル表現を獲得する．

1. 音源データから音響特徴量を計算する．
2. 音響特徴量を 250ms 毎のフレームに分割する．
3. 各フレームを音響特徴量のユークリッド距離に基づき k-means で k 個のクラスタに分割する．
4. 各フレームを割り当てられたクラスタで置き換え，word2vec を学習する．

最終的に，学習された各クラスタのベクトル表現をそのクラスタに含まれているフレームのベクトル表現として扱う．

音楽フレーズのベクトル化に用いる 5 つの素性を以下に示す．音響特徴量の計算には Python の音楽情報処理ライブラリ Librosa[4]を用いた．

MFCC(メル周波数ケプストラム係数)：
音色に関する特徴量で，人の知覚を考慮したスペクトルの概形を表す．

Chromagram：
和音に関する特徴量で，オクターブの 12 音について各音の成分の強さを表す．

Tempogram：
テンポに関する特徴量で，音のオンセット系列から求めた局所的なテンポ情報を表す．

Duration：
音楽フレーズの長さを表す特徴量である．楽曲全体の長さに対する相対的な長さを用いる．

Position：
楽曲中における音楽フレーズの位置を表す特徴量である．楽曲全体に対する絶対位置を用いる．

2.2.2 テキストフレーズのベクトル化

テキストフレーズのベクトルは以下の 3 つの素性を結合して表現される．

word2vec：
テキストフレーズの文脈に関する特徴量である．フレーズに含まれる各単語のベクトル表現の TF-IDF 値による重み付き平均を用いる．形態素解析には，MeCab[5]を使用し，単語ベクトルはあらかじ

[4]https://librosa.github.io/librosa/index.html
[5]https://taku910.github.io/mecab/

め学習された word2vec から獲得する．word2vec の学習，TF-IDF 値の計算には，Wikipedia のテキストを用いた．

Bag-of-Words：
テキストフレーズに含まれる音楽用語に関する特徴量である．あらかじめ用意しておいた音楽用語リスト中の各単語が，テキストフレーズ中に含まれていれば 1，それ以外は 0 となるようにベクトルを作成する．音楽用語リストは，「ハ長調」，「ff(フォルティッシモ)」，「コーダ」などの調や音量，楽曲を構成するフレーズに関する 115 単語から構成される．

Position：
解説文中におけるテキストフレーズの位置を表す特徴量である．テキストフレーズが属する解説文全体に対する絶対位置を用いる．

2.3 対応付け

ベクトルで表現された音楽フレーズ系列とテキストフレーズ系列を分類器とマッチングアルゴリズムを用いて対応付ける．まず，音楽フレーズとテキストフレーズがペアであるかどうかを識別する分類器を学習し，この分類器が出力するクラス確率を音楽フレーズとテキストフレーズの類似度スコアとする．時系列に並んだフレーズ系列同士をマッチングする手法には，動的計画法アルゴリズムの一つであるビタビアルゴリズム，または貪欲法を用いる．最後に，マッチング後の経路をヒューリスティックに補間して，多対多の対応を作成した結果を出力する．

2.3.1 分類器の獲得

音楽フレーズとテキストフレーズのペアを入力とし，「ペアである/ペアでない」の二値分類を行う分類器を学習する．分類器には，複数の決定木モデルを組合せることでアンサンブル学習を行う機械学習手法である Random Forest を用いる．「ペアである」クラスの所属確率を，音楽フレーズとテキストフレーズの類似度スコアとする．

2.3.2 マッチングアルゴリズム

音楽フレーズ系列とテキストフレーズ系列を対応付けるマッチングアルゴリズムには以下の 2 つを用いる．また，この対応付けは，音楽側を時間軸に取る場合 (music-to-text: m2t) とテキスト側を時間軸に取る場合 (text-to-music: t2m) の 2 つが考えられる．

ビタビアルゴリズム
最適な状態系列を求めるアルゴリズムで，動的計画法の一つである．本研究では，状態確率に分類器から得られる類似度スコアを使用し，遷移確率は定数とする．

貪欲法
各時刻で分類器の類似度スコアが最も高いフレーズを選択する．ただし，対応するフレーズの時刻が前後し，経路が作成されない場合を防ぐため，前の時刻で選択したフレーズ以降の中から選択する制約を与える．

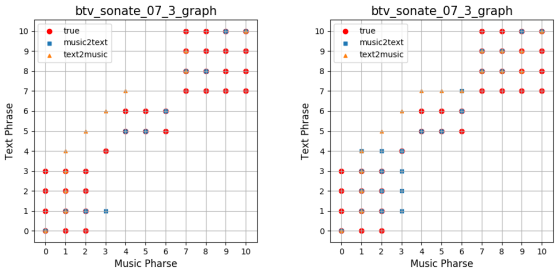

図 4: ビタビアルゴリズムによる音楽フレーズとテキストフレーズの対応付け結果の例 (左) と補間例 (右)

表 1: データセットに関する統計値

	件数	1 曲あたりの平均
楽曲	101	–
音楽フレーズ	2,813	27.9
テキストフレーズ	1,883	18.6
ペア	1,066	10.6

2.3.3 経路の補間

ビタビアルゴリズムによるマッチング結果と，その補間の例を図 4 に示す．マッチング結果の経路は，各時刻に対して一点のみを選択するため，対応するフレーズがない場合が存在する．そこで，すべてのフレーズについて対応するフレーズを選択するため，以下の方法で経路を補間する．

経路上の任意の異なる 2 点 (x_1, y_1) と (x_2, y_2) ($x_1 \leqq x_2$ かつ $y_1 \leqq y_2$) について，$y_2 - y_1 \geqq 2$ ならば，$x_1 \leqq i < x_2$ かつ $y_1 < j < y_2$ を満たす対応行列上の点 (i, j) をすべて選ぶ．

3 データセットの作成

ベートーヴェンのピアノソナタ楽曲の電子楽譜とその解説文を収集し，本研究の提案システムに用いるデータを作成した．データセットに関する統計値を表 1 に示す．電子楽譜は，MuseScore[6]で配布されているデータを収集した．解説文は，楽曲毎に解説が記載された「ベートーヴェン (作曲家別名曲解説ライブラリー)[3]」を用いた．

収集した電子楽譜と解説文データを，2.1 節の手法を用いて音楽フレーズとテキストフレーズに分割し，人手で電子楽譜上の各音楽フレーズを説明するテキストフレーズを選択することで，楽曲毎に音楽フレーズ系列とテキストフレーズ系列を対応付けたペアデータを作成した．

4 評価実験

第 3 節で作成したベートーヴェンのピアノ・ソナタ楽曲のデータセットを用いて，提案システムによる楽曲と解説文の対応付けの性能評価を行う．

[6]https://musescore.com/

表 2: 実験結果 (macro 平均値)

手法		Precision	Recall	F
ビタビアルゴリズム	m2t	0.352	0.425	**0.374**
	t2m	0.354	0.342	0.340
貪欲法	m2t	0.340	0.403	0.357
	t2m	0.352	0.326	0.332
ビタビアルゴリズム (補間前)	m2t	0.412	0.237	0.296
	t2m	0.380	0.156	0.215
貪欲法 (補間前)	m2t	0.396	0.226	0.283
	t2m	0.414	0.174	0.238
対角線 (ベースライン)		0.374	0.215	0.268

4.1 実験設定

音楽フレーズ素性に用いる音響特徴量のフレームベクトルに関して，クラスタリングのクラスタ数は 10,000 とした．また，フレームベクトルの word2vec の学習には収集したベートヴェンのピアノソナタ楽曲の音源データに加えて，ピアノ演奏データセット MAESTRO[6] の 1,184 の演奏音源データを使用し，ベクトルサイズは 100 次元とした．テキストフレーズ素性に用いる word2vec のベクトルサイズは 300 次元とした．Random Forest のハイパーパラメータは，自動最適化ツール Optuna[7]を用いて最適化を行った．

4.2 評価方法

101 曲の音楽フレーズとテキストフレーズのペアデータを用いて，対応付けの結果を Leave-One-Out で評価する．ベースラインとして，対応行列の対角線上の対応を正解とする手法を用いる．テストデータ 1 曲を選択し，残りの 100 曲を訓練データとする．訓練データの各楽曲について，正解ペアと同じ数の不正解ペアをランダムに作成し，正解/不正解の両ペアを分類器の学習に用いる．各テストデータ楽曲の対応付け結果に対する Precision，Recall，F 値の macro 平均を評価指標として用いる．

4.3 結果と考察

実験結果を表 2 に示す．ビタビアルゴリズム-m2t を用いた場合に，F 値が最も高かった．貪欲法よりスコアが高くなる理由として，貪欲法が各時刻のフレーズのみを考慮するのに対して，ビタビアルゴリズムは過去の状態を考慮した上で，経路のスコアを最大とするフレーズを選択することが考えられる．

また，m2t の手法の方が t2m よりも Recall が高くなった．これは，平均音楽フレーズ数が平均テキストフレーズ数の約 1.5 倍のため，音楽側を時間軸に取る方が補間される点が多いことが原因である．

補間前の経路はビタビアルゴリズム-m2t と貪欲法-m2t の両方が，対角線と比較して F 値が高くなった．このことから，あるテキストフレーズに対して，複数の音楽フレーズから対応を探す t2m の方が，m2t よりも問題として難しいことが分かる．原因としては，テキストフレーズに比べて音楽フレーズの数が多いことに加え，フレーズの繰り返しが多い，あるいは，全体が似たフレーズで構成されている楽曲では，音楽フレーズ同士の差を捉えるのが難しいことが考えられる．

精度が低い結果となった楽曲に関しては，音楽用語を使用していない，あるいは，参考資料や他曲の引用を含むテキストフレーズが存在する場合，テキスト側の曖昧性が高く，対応する音楽フレーズの探索が困難になることが原因として考えられる．また，楽曲内で主題の繰り返しが頻繁に出現するため，単一のフレーズ同士の比較以外に，前後複数のフレーズ情報や，繰り返しフレーズの位置など，大域的なフレーズ情報を活用することで，精度の向上が期待できる．

さらに，作成したデータセットが小さく，訓練データ中の出現頻度が少ない音楽用語を有効に活用できていないことが考えられ，データセットの拡張が今後の課題である．

[7]https://optuna.org/

5 おわりに

本研究では，楽曲と解説文の自動対応付けを行うシステムを提案した．ベートーヴェンのピアノソナタの楽曲から作成した音楽フレーズとテキストフレーズのペアデータを用いて，提案システムによる対応付けの精度の評価を行った結果，ベースラインより高い F 値が得られた．今後の課題としては，音楽フレーズ作成の自動化，対応付けの精度向上に向けてフレーズ情報を有効に活用する手法の検討，データセットの拡張が挙げられる．

謝辞 この成果は，JST さきがけ JPMJPR1655 の支援の結果得られたものです．

参考文献

[1] Ashwin K. Vijayakumar, Ramakrishna Vedantam, and Devi Parikh. Sound-word2vec: Learning word representations grounded in sounds. *EMNLP*, 2017.

[2] 高橋量衛, 大石康智, 武田一哉ほか. Web から収集した楽曲を説明するテキストと楽曲の音響特徴量との関連づけに関する検討. 情報処理学会研究報告音楽情報科学 (MUS), Vol. 2007, No. 102 (2007-MUS-072), pp. 43–48, 2007.

[3] 堀内久美雄. 作曲家別名曲ライブラリー ベートーヴェン. 音楽之友社, 1992.

[4] 亀岡弘和, 中村友彦, 高宗典玄. 音楽音響信号処理技術の最先端. 電子情報通信学会誌, Vol. 98, No. 6, pp. 467–474, 2015.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013.

[6] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the maestro dataset. *ICLR*, 2019.