歌詞のサビ区間検出手法

渡邉研斗　後藤真孝
産業技術総合研究所 (AIST)
{kento.watanabe, m.goto}@aist.go.jp

1　はじめに

ポピュラー音楽における「サビ」とは，楽曲中で最も繰り返され記憶に残る区間である[1]．音楽情報処理の分野では，音響信号に基づくサビ区間検出手法が活発に研究されてきたが[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1, 18, 19]，歌詞テキストに基づいた検出手法はこれまで提案されていない．

本研究の目的は，歌詞を入力したときに，そのサビ区間を自動的に検出することである．テキストのみからサビ区間を検出可能かどうかは自明ではなく，学術的な視点からも興味深い．更に，様々な検索システムで歌詞に基づくサビ検出技術が有用である．例えば，音楽視聴者が “I love you” などの特定のフレーズを含むサビを探したいとき，検索システムは歌詞のサビ区間の位置を知っている必要がある．

本研究では，英語・日本語の歌詞のサビ区間を検出する教師ありモデルを提案する．本モデルは，歌詞の繰り返しパターンを表す構造的特徴量と，サビ区間特有のフレーズを表す言語的特徴量の両方を考慮する．また，教師あり学習に必要となる訓練データを大量に用意するために，我々は歌唱時刻で同期された音響信号-歌詞のアライメントデータを利用し，1 万曲以上の歌詞に教師ラベルを自動付与した．実験では，提案特徴量の有効性や，自動付与した教師ラベルの有用性，英語と日本語のサビ区間に対する言語依存性など，様々な観点から検出タスクや歌詞のサビ区間の性質を調査した．

2　歌詞のサビ区間検出タスク

図 1 はサビ区間のラベルが付与された歌詞の例である．本タスクでは，入力歌詞には A メロ・B メロ・サビ区間の境界（空行）が一切存在しないと仮定し，全ての歌詞から空行を取り除いた．

本研究では，サビ区間検出タスクを系列ラベリング問題として定式化する．つまり，歌詞の各行の**サビ**および**非サビ**のラベルを予測する．X_s は T 行のテキスト $\{x_1, ..., x_t, ..., x_T\}$ で構成される曲 s の歌詞である．各行 x_t はバイナリラベル y_t を持つ．$y_t = 1$ のとき x_t はサビであり，$y_t = 0$ のとき x_t は非サビである．Y_s は X_s に対応するラベル系列 $\{y_1, ..., y_t, ..., y_T\}$ である．モデルの訓練では，条件付き確率 $P(Y_s|X_s)$ を学習する．モデルの評価では，訓練されたモデルは入力された行の系列 X_s のラベル系列 Y_s 予測する．

図 1 の例では，サビ区間は完全一致の繰り返しであるが，繰り返される行を抽出するだけではサビ区間の検出は困難である．例えば，図 1 の 9–12 行と 21–24 行は完全一致であるがサビ区間ではない．また，様々なバリエーションのサビ区間を検出するルールの作成も難しい．本研究では，様々な種類のサビ区間に対応するための特徴量を設計する．

3　歌詞のサビ区間のモデル化

本研究では，音響信号に基づくサビ区間検出手法で用いられる自己類似行列 (SSM) をテキストに応用することで，歌詞の繰り返しパターンを捉えた**構造的特徴量**を設計する．更に単語・文脈ベクトルを用いることで，サビ特有のフレーズを捉えた**言語的特徴量**を設計する．

以下の節では，まず歌詞の繰り返しパターンを表した SSM と，それらを構造的特徴量としてベクトル化する方法について説明する．次に word2vec[21] と context2vec[22] を用いて歌詞の意味的・統語的な情報をベクトル化することで得られる言語的特徴量について説明する．最後に，これら構造的・言語的特徴量を用いたニューラルネットワーク (NN) ベースの系列ラベリングモデルについて説明する．

3.1　構造的特徴量

これまで音響信号に基づく音楽解析の研究では，図 1 に示したような SSM を用いて繰り返される A メロやサビ区間を検出してきた．繰り返される区間

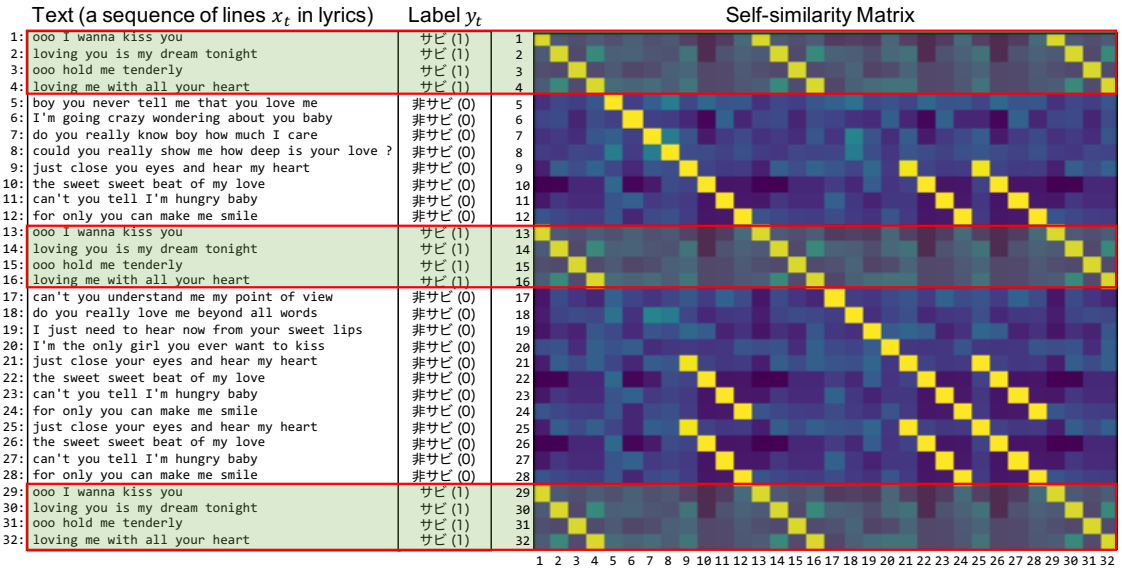

図 1 サビラベル付きの歌詞（RWC 研究用音楽 DB No.81[20]）と自己類似行列の例．各セルは行同士の類似度を表す．

は行列内で高い値を持つ斜線として表現され，この斜線のパターンが構造検出の手がかりとして使われてきた．本研究ではこの SSM による手法をテキストに応用することで，歌詞の繰り返しパターンを捉える．ただし，SSM 内の各セルの類似度計算によって，捉えられる繰り返しパターンが大きく異なるため，類似度の設計が重要となる．そこで本研究では，以下の 9 種類の類似度を設計する．

文字列類似度 行同士の標準化編集距離 [23].

先頭語類似度 各行の先頭単語間の標準化編集距離.

末尾語類似度 各行の末尾単語間の標準化編集距離.

発音類似度 押韻による繰り返しを捉えるために，発音記号の系列間の標準化編集距離を計算する．本研究では CMU 発音辞書を用いて英語歌詞の発音記号を抽出する．

品詞類似度 文法構造の類似度を計算するために，品詞系列間の標準化編集距離を計算する．本研究では NLTK の品詞解析器 [24] を用いる．

単語ベクトル類似度 行同士の意味的類似性を捉えるために，本研究では訓練済みの word2vec を用いて，各行内の単語ベクトルを平均し，それらのコサイン類似度を計算する．ただし，bag of words を仮定したこの類似度では語順の違いを区別できない．

文脈ベクトル類似度 語順を考慮するために，word2vec を LSTM を用いて拡張した context2vec を用いて，各行の単語列を LSTM でベクトル化し，コサイン類似度を計算する．

単語の音節数類似度 サビ区間の歌詞は，単語が全く異なっていても同じ音節数であることがあるため，各行で単語の音節数の系列を利用する．例えば，歌詞 *"Sometimes you lost yourself away"* と *"Everytime you just close your eyes"* の音節数の系列はそれぞれ $\{2,1,1,2,1\}$ と $\{2,1,1,1,1,1\}$ である．これらの音節数系列が類似している場合，繰り返しの可能性がある．本研究では動的時間伸縮法 (DTW)[25] を用いて，音節数系列間の類似度を計算する．

行の音節数類似度 本研究では各行内の全単語の合計音節数も使用する．例えば，図 1 に示されている全サビ区間では，最初の行の合計音節数は 6 であり，2 行目の合計音節数は 8 である．次の手順により，各行ペアの合計音節数の類似性を計算する．(1) 連続した 4 行 $L_t = \{x_t, x_{t+1}, x_{t+2}, x_{t+3}\}$ を抽出する．(2) 行 x_t と $x_{t'}$ 間の類似度を L_t と $L_{t'}$ の合計音節数の DTW によって計算する．

本研究で 9 種類の SSM を上記の類似度を用いて計算する．SSM は $\mathbf{A}_m \in \mathbb{R}^{T\times T}$ で表し，$m(1 \le m \le 9)$ は類似度の種類を意味する．次に，SSM から特徴量を計算するために，本研究では畳み込みニューラルネットワーク (CNN) を用いる（図 2）．各 SSM からターゲットとなる行を中心とした固定窓幅 w の部分行列を抽出する．ここで部分行列は $\mathbf{a}^t_m = \mathbf{A}_m[t-w+1,...,t+w; 1,...,T] \in \mathbb{R}^{2w\times T}$ で表す．CNN へ入力するのは 9 つの部分行列 $\{\mathbf{a}^t_{str},...,\mathbf{a}^t_{syL}\} \in \mathbb{R}^{2w\times T\times 9}$ であり，チャンネルの数は SSM の数に対応する．最初の 2D 畳み込み層のカーネルサイズは $(w+1)\times(w+1)$ であるため，SSM 内の

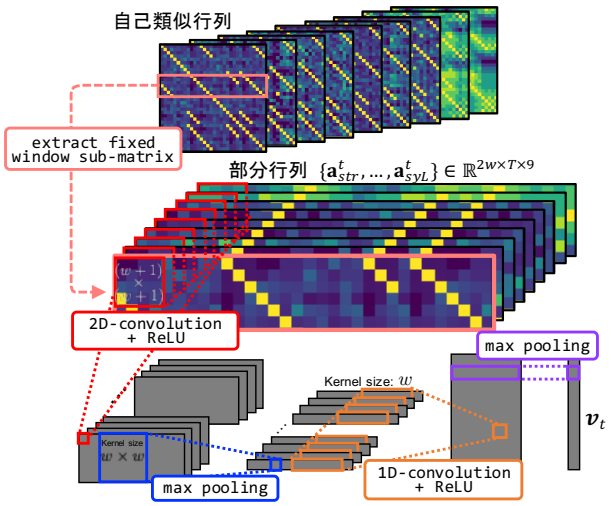

図 2　SSM のための畳み込みニューラルネットワーク．

斜線の端を捉えることができる．ここで得られたテンソルはカーネルサイズが $w \times w$ の max pooling によってダウンサンプルされる．次にカーネルサイズが w の 1D 畳み込み層を適用し，最後に max pooling によってダウンサンプルする．各畳込み層の活性化関数には ReLU を使用する．上記手順を歌詞の各行 x_t に実行し，構造的特徴量 $\boldsymbol{v}_t$ を計算する．

3.2　言語的特徴量

サビ区間に出現しやすい歌詞のフレーズを調査するため，サビ区間と非サビ区間の単語トライグラムの差を計算した．その結果，*"I'll"* や *"Let's"* などの未来に関するフレーズがサビ区間に頻出し，*"have been"* や *"didn't"* などの過去に関するフレーズが非サビ区間で頻出することがわかった．この傾向を利用するために，以下の特徴量を設計する．

単語ベクトルの平均 行内の単語を訓練済み word2vec を用いてベクトル化し，その平均を特徴量とする．

単語列のベクトル表現 訓練済み context2vec を用いて行をベクトル化したものを特徴量とする．

歌詞の各行 x_t について上記ベクトルを計算し，それらを連結した言語的特徴量 $\boldsymbol{u}_t$ を得る．

3.3　NN ベースの系列ラベリングモデル

本研究では，標準的な双方向 LSTM (Bi-LSTM)[26] を用いて，条件付き確率 $P(Y_s|X_s)$ を計算する．位置 t での Bi-LSTM 層への入力は構造的特徴量 $\boldsymbol{v}_t$ と，言語的特徴量 $\boldsymbol{u}_t$ の連結ベクトルである．条件付き確率 $P(Y_s|X_s)$ は softmax 関数を用いて計算される：

$$P(Y_s|X_s) = \frac{\exp(\mathrm{Score}(X_s, Y_s))}{\sum_{Y'_s} \exp(\mathrm{Score}(X_s, Y'_s))}. \quad (1)$$

ここで Score は以下の式で定義される．

$$\mathrm{Score}(X_s, Y_s) = \sum_t^T \mathrm{BN}(h_t[y_t]), \quad (2)$$

ここで $h_t[y_t]$ は位置 t の Bi-LSTM の出力であり，BN はバッチ正規化 [27] であり，ロス関数はクロスエントロピーを用いる．

4　実験

音響信号ベースのサビ検出の研究 [1] を参考にして，F-measure を用いて提案手法を評価した．ここで F-measure は $(2 \cdot R \cdot P)/(R + P)$ で計算され，各 R と P は以下の式で計算される．

$$P = \frac{\text{正しく検出されたサビ区間内の行数}}{\text{サビ区間として検出された行数}}.$$
$$R = \frac{\text{正しく検出されたサビ区間内の行数}}{\text{正解のサビ区間の行数}}.$$

更に，音楽構造解析の評価で広く用いられる Python パッケージ `mir_eval`[28] を用いて，pairwise F-measure (p-F)，normalized conditional entropy F-measure (n-F)，V-measure を計算した．

4.1　モデルパラメータ

窓幅は 3 とし，2D および 1D 畳み込み層のカーネル数をそれぞれ 200 と 400 とした．Bi-LSTM の隠れ層の次元を 600 とした．word2vec と context2vec の次元数は 300 として，歌詞データを用いて事前訓練した．パラメータの最適化には AdamW[29] を使用し，学習率は 0.001，バッチサイズは 64 とした．訓練は 100 エポック行い，開発セットにおける最も高い F-measure であるエポックのモデルを評価に用いた．

4.2　データセット

各行がサビかどうかを予測する教師ありモデルを訓練するためには，図 1 に示したような行単位で教師ラベルを持つ大量の歌詞データが必要となる．本研究では以下の手順によって歌詞データの各行に教師ラベルを自動付与した．(1) 我々は歌唱時刻で同期された音響信号-歌詞のアライメントデータを 100,772 曲だけ用意した．(2) 音響信号ベースのサビ区間検出手法 [1] を用いて，サビ区間の開始時刻と終了時刻を検出した．(3) 音響信号から検出されたサビ区間内に存在する歌詞の行にサビラベルを付与した．本研究では 9,313 曲の英語歌詞と，91,459 曲の日本語歌詞に教師ラベルを自動付与し，それぞれのデータを EN_auto と JA_auto と呼ぶ．

パラメータ調整や評価のために，信頼性の高い教

表 1　実験結果：構造的・言語的特徴量の重要性.

特徴量	訓練データ / テストデータ							
	EN_auto / EN_test				JA_auto / JA_test			
	F	*p-F*	*n-F*	*V*	*F*	*p-F*	*n-F*	*V*
構造的特徴量	77.9	76.1	48.6	45.5	81.2	82.7	63.6	59.6
言語的特徴量	57.4	59.9	16.5	6.9	55.2	61.8	22.1	16.7
両方の特徴量	**78.1**	**77.7**	**50.8**	**47.3**	**83.4**	**83.5**	**64.9**	**61.4**

表 2　実験結果：自動付与した教師ラベルの信頼性.

訓練データ	*F*	*p-F*	*n-F*	*V*
JA_auto (91,459 曲)	**83.4**	**83.5**	**64.9**	**61.4**
JA_man (1,103 曲)	80.3	77.3	53.3	50.4

師ラベルを持つ 3 つの歌詞データを用意した.

(a) 訓練の比較用データ自動付与した教師ラベルが訓練において信頼できるか検証するために，比較用訓練データとして 1,103 曲の日本語歌詞に教師ラベルを手動で付与した．このデータを JA_man と呼ぶ.

(b) モデルパラメータの調整用データ我々は RWC 研究用音楽データベースの英語 21 曲と日本語 79 曲の歌詞にラベルを手動で付与し，このデータをモデルパラメータの調整用に使用した.

(c) テストデータ 118 曲の英語歌詞と 128 曲の日本語歌詞にラベルを手動で付与し，それぞれを EN_test と JA_test と呼ぶ．これらはサビ区間検出手法の評価のために使用した.

4.3　構造的・言語的特徴量の重要性

構造的・言語的特徴量の有効性を調査するために，各特徴量を用いたモデルの性能を比較した．表 1 より，構造的特徴量のみを使ったモデルは，言語的特徴量のみを使ったモデルよりも大幅に優れることがわかった．また，両特徴量を使用することで性能が更に向上した．これらの結果は，音響信号のサビ区間検出で用いられる SSM を歌詞に応用することの重要性を示すだけでなく，言語的特徴量の追加が歌詞のサビ区間検出に役立つことを示している.

4.4　自動付与した教師ラベルの信頼性

教師ラベルを大量に自動付与した JA_auto と，教師ラベルを少量だが手動付与した JA_man の訓練データとしての性能を比較をする．表 2 より，JA_auto で訓練したモデルが JA_man で訓練したモデルよりも高性能であることがわかる．この結果は，教師ラベルが自動的に付与されたものであっても，十分なデータサイズであればモデルの訓練において十分な信頼性を持つことを意味する.

表 3　実験結果：訓練データサイズと言語依存性.

訓練データ	テストデータ	*F*	*p-F*	*n-F*	*V*
EN_auto (9,313 曲)	EN_test	77.9	76.1	48.6	45.5
JA_auto (91,459 曲)	EN_test	80.3	80.6	58.1	54.4
EJ_auto (100,772 曲)	EN_test	**81.0**	**82.3**	**60.7**	**57.4**

4.5　訓練データサイズと言語依存性

表 1 より，英語モデルよりも日本語モデルの方が性能が良いことがわかるが，これは訓練データの量が大きく異なるためだと考えられる．そこで本研究では，大量の日本語データで訓練されたモデルが，英語のサビ区間をより正確に検出できるかどうか調査した．なお本実験では，構造的特徴量のみを考慮したモデル[1)]を用いる．表 3 より，日本語データで訓練されたモデルが，英語データで訓練されたモデルよりも，英語のサビ区間を検出できることがわかった．更に，英語と日本語をあわせたデータ (EJ_auto) で訓練されたモデルの性能が最も優れていることがわかった．これらの結果は (1) 異なる言語で訓練されたモデルでもサビ区間を検出できること，(2) 歌詞の繰り返しパターンは言語に依存しないこと，(3) 異なる言語データを混合することでサビ区間の検出性能が向上することを意味する．これは，少リソースの言語データであっても，利用可能な他言語リソースと混ぜることで，サビ区間が検出できる可能性があることを意味する.

5　おわりに

本論文は楽曲のサビ区間を歌詞のみから検出するという新しいタスクと，その手法を提案した．本研究の貢献は以下である：(1) サビ区間の構造的及び言語的特性を捉えるために，様々な特徴量を設計した．(2) 歌詞のサビ区間を検出する系列ラベリングモデルを提案した．(3) サビ区間の注釈付きの大規模な訓練データセットを作成する手法を示した．(4) 特徴量の重要性，訓練データの量，言語依存性などの様々な観点から，検出タスクや歌詞のサビ区間の性質を調査した．今後は A メロや B メロなどの区間も検出できるように手法を拡張する.

謝辞　本研究は，RWC 研究用音楽データベースと，株式会社シンクパワーから提供された歌詞データを利用した．また，本研究は JST ACCEL (JPMJAC1602) および科研費 (20K19878) の支援を受けた.

1)　SSM は繰り返しのパターンを表した行列に過ぎないため，異なる言語であってもモデルへの入力が実装上可能である.

参考文献

[1] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 5, pp. 1783–1794, 2006.

[2] Go Shibata, Ryo Nishikimi, Eita Nakamura, and Kazuyoshi Yoshii. Statistical music structure analysis based on a homogeneity-, repetitiveness-, and regularity-aware hierarchical hidden semi-markov model. In *Proceedings of ISMIR 2019*, pp. 268–275, 2019.

[3] Akira Maezawa. Music boundary detection based on a hybrid deep model of novelty, homogeneity, repetition and duration. In *Proceedings of IEEE ICASSP 2019*, pp. 206–210, 2019.

[4] Gabriel Sargent, Frédéric Bimbot, and Emmanuel Vincent. Estimating the structural segmentation of popular music pieces under regularity constraints. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 25, No. 2, pp. 344–358, 2017.

[5] Tian Cheng, Jordan B. L. Smith, and Masataka Goto. Music structure boundary detection and labelling by a deconvolution of path-enhanced self-similarity matrix. In *Proceedings of IEEE ICASSP 2018*, pp. 106–110, 2018.

[6] Jordan B. L. Smith and Masataka Goto. Using priors to improve estimates of music structure. In *Proceedings of ISMIR 2016*, pp. 554–560, 2016.

[7] Thomas Grill and Jan Schlüter. Music boundary detection using neural networks on combined features and two-level annotations. In *Proceedings of ISMIR 2015*, pp. 531–537, 2015.

[8] Brian McFee and Dan Ellis. Analyzing song structure with spectral clustering. In *Proceedings of ISMIR 2014*, pp. 405–410, 2014.

[9] Geoffroy Peeters and Victor Bisot. Improving music structure segmentation using lag-priors. In *Proceedings of ISMIR 2014*, pp. 337–342, 2014.

[10] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller. Converting path structures into block structures using eigenvalue decompositions of self-similarity matrices. In *Proceedings of ISMIR 2013*, pp. 209–214, 2013.

[11] Oriol Nieto and Tristan Jehan. Convex non-negative matrix factorization for automatic music structure identification. In *Proceedings of IEEE ICASSP 2013*, pp. 236–240, 2013.

[12] Florian Kaiser and Geoffroy Peeters. A simple fusion method of state and sequence segmentation for music structure discovery. In *Proceedings of ISMIR 2013*, pp. 257–262, 2013.

[13] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluís Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proceedings of AAAI 2012*, 2012.

[14] Meinard Müller, Peter Grosche, and Nanzhu Jiang. A segment-based fitness measure for capturing repetitive structures of music recordings. In *Proceedings of ISMIR 2011*, pp. 615–620, 2011.

[15] Jouni Paulus, Meinard Müller, and Anssi Klapuri. State of the art report: Audio-based music structure analysis. In *Proceedings of ISMIR 2010*, pp. 625–636, 2010.

[16] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 6, pp. 1159–1170, 2009.

[17] Meinard Müller and Sebastian Ewert. Joint structure analysis with applications to music annotation and synchronization. In *Proceedings of ISMIR 2008*, pp. 389–394, 2008.

[18] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In *Proceedings of WASPAA 2003*, pp. 127–130, 2003.

[19] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE ICME 2000*, p. 452, 2000.

[20] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC Music Database: Popular, classical and jazz music databases. In *Proceedings of ISMIR 2002*, Vol. 2, pp. 287–288, 2002.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS 2013*, pp. 3111–3119, 2013.

[22] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of SIGNLL 2016*, pp. 51–61, 2016.

[23] Yujian Li and Bi Liu. A normalized Levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 1091–1095, 2007.

[24] Steven Bird. NLTK: the natural language toolkit. In *Proceedings of ACL 2006*, 2006.

[25] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of Workshop on Knowledge Discovery in Databases*, pp. 359–370, 1994.

[26] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of IEEE ICASSP 2013*, pp. 6645–6649, 2013.

[27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of ICML 2015*, Vol. 37, pp. 448–456, 2015.

[28] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of ISMIR 2014*, pp. 367–372, 2014.

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of ICLR 2019*, 2019.