メロディ条件付き歌詞言語モデル

渡邉 研斗 † 松林 優一郎 † 深山 覚 ‡ 乾 健太郎 †,◇ 後藤 真孝 ‡ 中野 倫靖 ‡
東北大学 † 産業技術総合研究所 ‡ 理研 AIP◇
{kento.w, y-matsu, inui}@ecei.tohoku.ac.jp,
{s.fukayama, m.goto, t.nakano}@aist.go.jp

1 はじめに

メロディに対して歌詞を作る作業は，一般的な散文を作る作業と比べ，言語的な要素とメロディの音符・休符やリズム，繰り返しなどの音楽的な要素の両方を考慮する必要があるため，難易度の高い作業である [1]．例えば図1の上の歌詞のように，休符をまたいで単語を配置すると不自然な歌詞になるため，図1の下の歌詞のように休符をまたがないよう単語を配置する必要がある．

作詞作業において，仮に，歌詞の内容と上記のような音楽的な制約を同時に考慮して単語列を探索するシステムが実現できれば，膨大な可能性から単語列を探す手間を簡略化できると考えられる．このような動機から，歌詞の自動生成に関する研究が行われてきた [2–4]．例えば，Oliveira らはメロディの拍子と歌詞のアクセントの関係性をモデル化することで，入力メロディに対してアクセントの強弱が自然な歌詞を生成している．しかし，これらの研究ではメロディと歌詞の音韻的な側面しかモデル化していないため，入力メロディに対して短いフレーズの歌詞しか生成できない．

本研究の目的は，1 曲分の入力メロディに対して自然な位置に単語・文・段落の境界を配置した流暢な単語列を生成する新しい歌詞言語モデルの構築である．この目的の達成のために次の 3 つの課題に取り組む．(1)Web 上で公開されている電子楽譜データを収集し，対応する歌詞のテキストをアラインすることで，言語モデルの学習や分析に必要な，メロディに歌詞とその文・段落構造が付与された楽曲データを作成する．(2) 作成したデータを用いて，メロディと歌詞の文・段落構造の相関を分析する．(3) 分析結果から得られた知見を活かし，メロディに対して自然な文・段落境界を持ち，かつ日本語文として流暢な歌詞を生成する新しい言語モデルを構築する．具体的には標準的な Recurrent Neural Network Language Model (RNNLM) [5] を拡張し，RNN 層の各時刻に，直前に生成された語の他に音符の種類や長さを特徴量としたベクトルを入力する．

実験の結果，我々の提案する歌詞言語モデルが，入力メロディと文・段落構造の関係性を捉えながら流暢な歌詞を自動生成できることが確かめられた．

2 メロディ–歌詞対応データの作成

メロディと歌詞の関係性をモデル化するために，メロディに対して歌詞の文字（モーラ）・単語・文・段落の情報が対応づいた楽曲データが必要である（図2の下段）．我々は Web から収集可能な電子楽譜データ（図2の上段）と歌詞のテキストデータ（図2の中段）を組み合わせることで目的のデータを作成する．

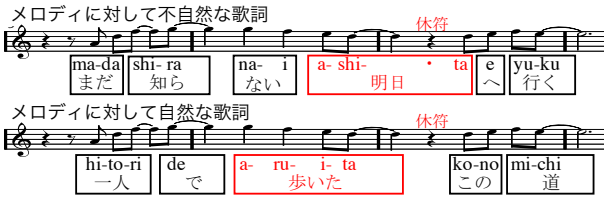

図1 メロディに対して不自然な歌詞と自然な歌詞．この例では RWC 研究用音楽 DB(No.20) を用いた [6]．

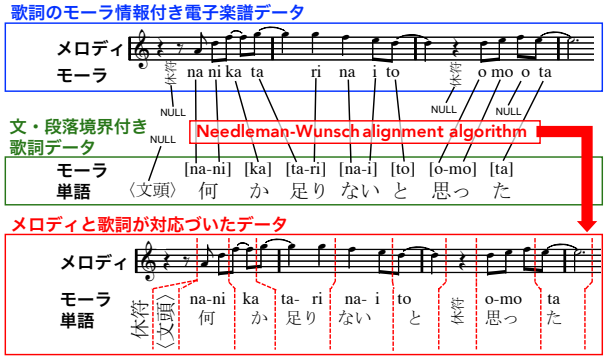

図2 メロディと歌詞の自動アライメント

我々はまず，メロディの各音符に歌詞のモーラ情報が付与されている電子楽譜データを収集した．しかし，このデータは歌唱するために十分な情報は含んでいるものの，文や段落の情報は含まれていないため，メロディと歌詞の構造をモデル化する目的には不十分である．そこで，文と段落の境界が記された歌詞データを別途収集し，以下の手順により電子楽譜データに歌詞の単語・文・段落情報を付与した．(1) まず，文・段落境界が記された歌詞データに対して形態素解析を行い，単語の境界とモーラ数を推定した．(2) 次に，Needleman-Wunsch アルゴリズム [7] を用いて，楽譜データと歌詞データをモーラ単位で読みが一致するよう j にアライメントを行った．本手順によりメロディと歌詞が対応づいた 1,000 曲のデータを作成した．

3 メロディと歌詞の相関分析

作詞家は歌いやすい歌詞を作成するために，音符や休符を考慮しながら単語を選択すると言われている [1]．例えば，図3のように短い休符＃ 2 の直後には単語境界

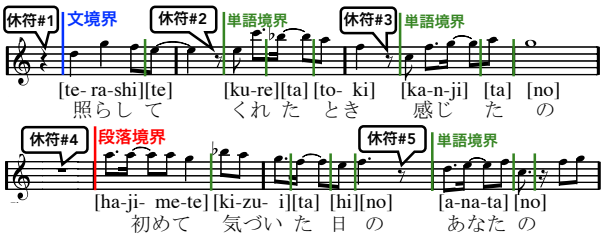

図3　休符の直後に出現する歌詞の境界の例

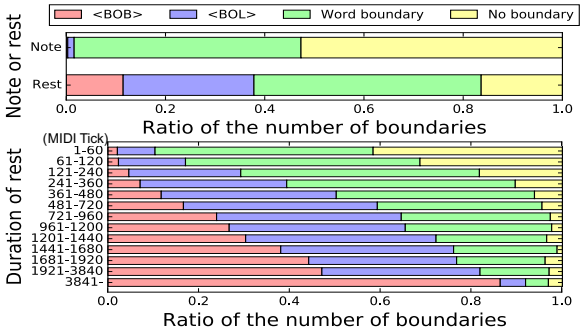

図4　メロディ中の歌詞の単語・文・段落境界の出現割合

が現れ，長い休符＃ 4 の直後には段落境界が現れる．本研究では，「歌詞の単語・文・段落の境界の出現位置はメロディ中の休符の直後に発生しやすく，また休符の長さが長いほど出現しやすい」という仮説を立て，作成した楽曲データを定量的に分析することで，この仮説を検証する．

まず，休符直後に対する文・段落境界の表れやすさを検証する．図4の上に，音符・休符の直後における各境界の出現割合を示した．この図において ⟨BOL⟩ と ⟨BOB⟩ はそれぞれ文境界 (Beginning of Line) と段落境界 (Beginning of Block) を表す．この図より，文・段落境界は音符の直後には殆ど出現することはなく，概ね休符の直後に現れることがわかる．次に，図4の下に休符の長さ別の単語・文・段落境界の分布を示す．ここで 480 と 1980 はそれぞれ四分休符と全休符を表す．この図から，休符の長さが長くなるにつれて，単語・文・段落境界の順に出現しやすくなることがわかる．

以上の結果より，休符の長さによって文・段落境界の出現分布が偏るということが確かめられた．ここで注目すべき点は，休符の直後に必ず歌詞の境界が出現するわけではなく，休符の長さに応じて確率的に各境界が現れるということである．今回得られた知見は，確率的手法のよってメロディと歌詞の関係性をモデル化できることを示唆している．このような知見から，我々は確率的手法によってメロディと歌詞の関係性をモデル化する．

4　メロディ条件付き言語モデル

本節では，メロディに対して自然な文・段落境界を持ち，かつ流暢な歌詞を生成する「メロディ条件付き RNNLM」を提案する．提案モデルのネットワーク構造を図5に示す．モデルの入力メロディとして，音符もしくは休符の系列 $\mathbf{m} = m_1,...,m_i,...,m_I$（各 m は音の高さと長さの情報を持つ）が与えられる．そして各時

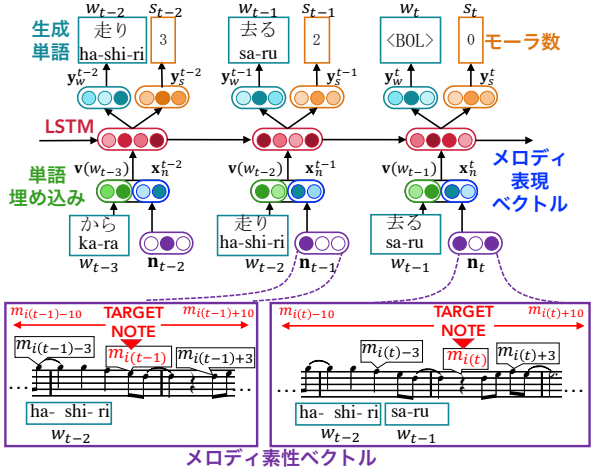

図5　メロディ条件付き RNNLM

刻 t において「直前に生成した単語 w_{t-1}」と「時刻 t 周辺のメロディを素性化したベクトル $\mathbf{n}_t$」をモデルに入力し，単語 w_t を生成する．この出力 w_t の候補には，文・段落境界を表す ⟨BOL⟩ と ⟨BOB⟩ も含まれており，モデルはメロディに対して文・段落境界を含んだ 1 曲分の歌詞を生成する．なお，提案モデルは直前に生成した単語のモーラ数から，次に入力する音符の位置 i を計算する必要がある．本研究では，入力したメロディの音符と歌詞のモーラは 1 対 1 に対応していると仮定することで，単語 w_t から音符の位置 i を一意に計算できるようにした．具体的には，単語の位置 t から音符の位置 i を返す関数 $i(t)$ を用意した．

ここで技術的な課題として，休符の位置と歌詞のモーラ数の関係性のモデル化が挙げられる．例えば，図5でモデルに入力したメロディ $m_{i(t-1)}, m_{i(t)}, ...$ が「音符，音符，長い休符，音符，音符」のとき，休符をまたがないように，モーラ数が 2 以下の単語を生成することが望ましい．もし，十分な量の訓練データがあれば，入力メロディに対して自然な単語を生成するモデルが学習されると考えられる．しかし，学習に使えるメロディ・歌詞データは 1,000 曲しかなく学習には十分ではない．

本研究では，この課題を解決する 2 つのアプローチを試みる．まず小規模なメロディ・歌詞データと大規模な歌詞のテキストデータを組み合わせた学習戦略を提案する．次に，休符の位置と単語のモーラ数の関係性を学習するために，単語 w_t と単語のモーラ数 s_t を同時に予測するモデルを構築する．

4.1　モデル構成

提案モデルは標準的な RNNLM [5] を基本とする：

$$P(\mathbf{w}) = \prod_{t=1}^{T} P(w_t | w_0, ..., w_{t-1}) \qquad (1)$$

ここで入力単語 $w_0, ..., w_{t-1}$ は LSTM を用いてエンコードされ，softmax 関数によって単語 w_t の生成確率を計算する．なお $w_0 = \langle \mathrm{B} \rangle$ は歌詞の先頭を表す．我々はこの RNNLM を拡張し，メロディを素性化したベクトルの系列 $\mathbf{n}_1, ..., \mathbf{n}_t$ を条件部に加える：

$$P(\mathbf{w}|\mathbf{m}) = \prod_{t=1}^{T} P(w_t | w_0, ..., w_{t-1}, \mathbf{n}_1, ..., \mathbf{n}_t) \qquad (2)$$

さらに，LSTM パラメータを共有しながらモーラ数

も同時に予測する．この拡張により，休符の位置とモーラ数の対応性を明示的に学習することができる：

$$P(\mathbf{s}|\mathbf{m}) = \prod_{t=1}^{T} P(s_t|w_0, ..., w_{t-1}, \mathbf{n}_1, ..., \mathbf{n}_t) \quad (3)$$

ここで $\mathbf{s} = s_1, ..., s_T$ はモーラ数の系列であり生成単語列 $\mathbf{w} = w_1, ..., w_T$ に対応している．

各時刻 t において，softmax 活性化関数により単語分布 $\mathbf{y}_w^t \in \mathbb{R}^V$ とモーラ数分布 $\mathbf{y}_s^t \in \mathbb{R}^S$ を出力する：

$$\mathbf{y}_w^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_w \mathbf{z}_t)) \quad (4)$$

$$\mathbf{y}_s^t = \mathrm{softmax}(\mathrm{BN}(\mathbf{W}_s \mathbf{z}_t)) \quad (5)$$

ここで $\mathbf{z}_t$ は各時刻の LSTM の出力を表す．V は語彙数，S は最大モーラ数を表す，$\mathbf{W}_w$ と $\mathbf{W}_s$ は重み行列であり，BN は Batch Normalization [8] を表す．

直前に生成した単語の埋め込みベクトル $\mathbf{v}(w_{t-1})$ と，メロディ素性ベクトル $\mathbf{n}_t$ を非線形変換したベクトル $\mathbf{x}_n^t$ を連結したベクトル $\mathbf{x}^t$ を LSTM に入力する：

$$\mathbf{x}^t = [\mathbf{v}(w_{t-1}), \mathbf{x}_n^t] \quad (6)$$

$$\mathbf{x}_n^t = \mathrm{ReLU}(\mathbf{W}_n \mathbf{n}_t + \mathbf{b}_n) \quad (7)$$

ここで $\mathbf{W}_n$ は重み行列，$\mathbf{b}_n$ はバイアスである．

歌詞生成タスクでは，生成確率値（式 (2, 3)）が大きい単語列をビーム探索する．生成単語列の合計モーラ数が入力メロディの音符数に達したら，生成を止める．

4.2 メロディ素性ベクトル

時刻 t 周辺の音符・休符の系列を抽出し，それらの情報をメロディ素性ベクトル $\mathbf{n}_t$ としてエンコードし，モデルに入力する（図5）．ここで，メロディ素性ベクトル $\mathbf{n}_t$ は音符と休符の高さと長さを表す素性ベクトルで表現される．具体的には，音符 $m_{i(t)}$ を中心とした前後 10 個の音符系列 $m_{i(t)-10}, ..., m_{i(t)}, ..., m_{i(t)+10}$ の音符と休符の高さと長さを計算し，それらの情報に対応する場所の値が 1 であるバイナリベクトル $\mathbf{n}_t$ がモデルに入力される．

4.3 学習戦略

事前学習　提案モデルの学習に必要なメロディと歌詞が対応づいたデータは 1,000 曲しかない．一方，大規模な歌詞のテキストデータを用意することができる．そこで 54,081 曲の歌詞データで提案モデルを事前学習した後に，学習されたパラメータを引き継いだ状態でメロディと歌詞が対応づいたデータを用いてモデルを再学習をする．なお事前学習では，メロディ素性ベクトル $\mathbf{n}_t$ には値が 0 のベクトルを用いる．この学習戦略で得られたモデルを *Fine-tuned* モデルと呼ぶ．

擬似メロディを用いた学習　歌詞のテキストデータに擬似メロディを割り当てることでメロディ・歌詞データを 54,081 曲まで増やす．この擬似データを用いて学習したモデルを *Pseudo-melody* モデルと呼ぶ．

擬似メロディの生成手順をアルゴリズム1に示した．まず，入力歌詞の各モーラに対して音符を割り当てる．次に，割り当てた音符の直後に休符を配置するか決定する．ここで，単語・文・段落境界を条件とした音・休符の生成確率を用いるが，3節の分析結果より境界と休符の統計的傾向が既知のため，各確率の計算が可能である．

Algorithm 1 擬似メロディ生成

```
for each mora in the input-lyrics do
    b ← get boundary type next to the mora
    sample 音符の高さ p ~ P(p_i | p_{i-2}, p_{i-1})
    sample 音符の長さ d_note ~ P(d_note | b)
    assign 音符 with (p, d_note) to the mora
    sample binary variable r ~ P(r | b)
    if r = 1 then
        insert 休符 with 長さ d_rest ~ P(d_rest | b)
    end if
end for
```

5 実験

提案モデルを 2 つの評価指標を用いて評価する．

テストセットパープレキシティ (PPL)　PPL は言語モデルの標準的な評価指標である．PPL が小さいモデルほどテストセットの単語を予測できることを意味しているため，PPL はモデルが流暢な文を生成できるかどうかの性能評価として用いられる．なお，単語の予測性能だけを評価するために，⟨BOL⟩ と ⟨BOB⟩ を予測単語から除外した PPL-W も評価指標として設計した．

文・段落境界の生成性能　人間が作成した歌詞の文や段落の境界は，メロディに対して自然な位置に配置されているはずである．そこで，提案モデルが生成した ⟨BOL⟩ と ⟨BOB⟩ の位置が人間が作詞した歌詞の ⟨BOL⟩ と ⟨BOB⟩ の位置とどれだけ一致しているか計算することで，モデルが自然な位置に文・段落境界を生成できるか，F_1 値を用いて評価する．

5.1 設定

パラメータ　単語埋め込みとメロディベクトル $\mathbf{n}_t$ の次元数をそれぞれ 512 と 256 とし，LSTM の内部状態ベクトルの次元数は 768 とした．出力 $\mathbf{y}_w^t$ と $\mathbf{y}_s^t$ の loss 関数はクロスエントロピーを用いた．最適化アルゴリズムは Adam を用い，学習率は 0.001 とした．ミニバッチ数は 32 とし，early-stopping を用いた．

データ　訓練データとして，54,081 曲の日本のポピュラー音楽の歌詞を用い，その内 900 曲はメロディ・歌詞データである．これらとは別の 100 曲のメロディ・歌詞データをテストセットとする．また，学習対象の語彙はモーラ数が 10 以下の出現頻度が大きい上位 2 万の単語を使用し，その他の単語は未知語タグに置換する．

5.2 メロディ条件付き言語モデルの効果

提案モデルの効果を調査するために，以下の 4 モデルを比較する．(1)*Lyrics-only* モデル：54,081 曲の歌詞だけを学習した RNNLM．(2)*Alignment-only* モデル：900 曲のメロディ・歌詞データのみで学習した提案モデル．(3)*Fine-tuned* モデル，(4)*Pseudo-melody* モデル：提案した学習戦略を用いたモデル．

表1に実験結果を示す．この表より文・段落境界の生成性能に関しては，提案した *Alignment-only*, *Fine-tuned*, *Pseudo-melody* モデルが高精度であり，提案手法がメロディと歌詞の関係性を捉えていることがわかった．特に，文境界の再現は *Fine-tuned* モデルが，段落境界の再現は *Pseudo-melody* モデルが高性能となった．

PPL に関しては，大量のテキストデータを用いた *Lyrics-only*/*Full-data*/*Pseudo-melody* モデルが高性

Model	PPL	PPL-W	⟨BOB⟩F_1	⟨BOL⟩F_1	UB F_1
Lyrics-only	138.0	225.0	0.121	0.061	0.106
Alignment-only	173.3	314.8	0.298	0.287	0.477
Fine-tuned	152.2	275.5	0.260	**0.302**	**0.479**
Pseudo-melody	**115.7**	**197.5**	**0.318**	0.241	0.406

表1　メロディ条件付き言語モデルの効果．UB は ⟨BOB⟩ と ⟨BOL⟩ を区別しない場合を表す．

Model	PPL	PPL-W	F_1-UB
Fine-tuned	**152.2**	**275.5**	**0.479**
Fine-tuned (w/o $\mathbf{y}_s$)	155.1	278.1	0.366
Pseudo-melody	**115.7**	**197.5**	**0.406**
Pseudo-melody (w/o $\mathbf{y}_s$)	118.0	201.5	**0.406**

表2　モーラ数予測層の効果．(w/o $\mathbf{y}_s$) はモーラ数予測層 $\mathbf{y}_s$ の除外を表す．

能であるとわかる．一方，事前学習戦略を用いた *Fine-tuned* モデルは *Lyrics-only* モデルと比較して PPL の性能が悪い．これは，少ない楽曲データで追加学習し，過学習が起こったためと考えられる．同様に *Alignment-only* モデルも歌詞の言語モデルを学習するにはデータサイズが不十分である結果となった．興味深いことに，*Pseudo-melody* モデルの PPL が最高性能であることがわかる．この結果は，*Pseudo-melody* モデルが歌詞の単語列を予測するのに，入力されたメロディの情報を活用していることを意味する．

5.3　モーラ数予測層の効果

単語の予測層 $\mathbf{y}_w$ の学習と同時にモーラ数の予測層 $\mathbf{y}_s$ を学習することで，モデルの性能にどのような効果をもたらしたか調査するために，モーラ数予測層 $\mathbf{y}_s$ を取り除いたモデルと提案モデルを比較する．表2に比較結果を示す．少量の楽曲データを学習した *Fine-tuned* モデルにおいては，モーラ数予測層も同時に学習することで性能が改善していることがわかる．これは，メロディと歌詞が対応づいたデータが少量であっても，入力メロディに対してモーラ数を予測するようにモデルを同時学習することで，データスパースネス問題が軽減されることを意味する．一方で，大量の擬似楽曲データを学習した *Pseudo-melody* モデルにおいて，モーラ予測層の有無による性能差はないことがわかる．これは，提案モデルがメロディに対して自然な単語を生成するための訓練データ量が十分であることを意味する．

5.4　クラウドソーシングによる歌詞の主観評価

生成された歌詞に対して人がどのような印象を持つのか，クラウドソーシングを用いて主観評価する．本評価では，評価者は歌詞をメロディと一緒に聴き，歌詞の「聴きやすさ」「文法的な正しさ」「文単位で意味がわかるか」「曲全体で意味がわかるか」「総合評価」について 5 段階評価（1：悪い〜 5：良い）する．評価セットとして，RWC 研究陽音楽データベース [6] からランダムに 4 曲のメロディを選んだ．次に，各メロディに対して，生成手法が異なる 4 パターンの歌詞を生成する．具体的には，3 つの言語モデル (*Lyrics-only, Fine-tuned, Pseudo-melody*) による生成歌詞と (4) 人間が作詞した歌詞を評価に使用する．評価に使用した歌詞は公開中である[*1]．本評価では，各楽曲に対して 50 人に評価をしてもらい，計 200 サンプルを収集した．なお，評価者は歌詞が自動生成されたことを知らされていない．

[*1] `http://www.cl.ecei.tohoku.ac.jp/lyrics`

評価項目	*Lyrics-only* 平均 ± SD	*Fine-tuned* 平均 ± SD	*Pseudo-melody* 平均 ± SD	*Human* 平均 ± SD
聴きやすさ	2.10±1.21	**2.98**±1.03	**3.10**±1.02	3.86±0.96
文法	**2.93**±1.07	2.68±1.03	**3.03**±0.99	3.80±0.94
意味（文単位）	**2.92**±1.00	2.55±1.02	**2.94**±0.99	3.80±0.98
意味（曲全体）	**2.72**±0.96	2.45±1.00	**2.72**±0.99	3.76±1.05
総合評価	2.39±0.96	2.19±0.99	**2.47**±0.93	3.32±1.14

表3　クラウドソーシングによる 5 段階評価の結果

表3に各手法が生成した歌詞に関する 5 段階評価結果の平均と標準偏差 (SD) を示した．「聴きやすさ」の評価項目に関しては，メロディと歌詞を学習した *Fine-tuned* モデルと *Pseudo-melody* モデルが高評価となり，表1の境界生成性能と同様の結果となった．一方，「文法」と「意味」の評価項目に関しては，*Lyrics-only* モデルと *Pseudo-melody* モデルが高評価となり，表1の PPL の性能と同様の結果となった．総合的な評価に関しては，*Pseudo-melody* が 3 モデルの中で最も高い評価結果となった．これらの結果は，擬似データの学習戦略を用いることが質の良い歌詞生成に寄与していることを意味する．しかし，生成された歌詞は人間が作詞した歌詞の質には及ばず，生成手法に改善の余地がある．

6　おわりに

本研究では，入力メロディに対して自然な位置に文と段落を配置し，かつ流暢な歌詞を生成する言語モデルを新たに提案した．モデル構築の前準備として，メロディと歌詞が対応づいた楽曲データを 1,000 曲作成し，メロディと歌詞の文・段落構造の相関を分析した．実験の結果，以下の結論が得られた．(1) 提案モデルは流暢な単語列の生成能力を持ちながら，長い休符の直後に文・段落境界を持つ歌詞を生成することができる．(2) 少量の楽曲データと大規模なテキストデータを組み合わせて学習することで，モデルの性能が向上する．(3) 擬似メロディを学習したモデルが生成した歌詞が，主観評価において高評価であった．今後は，メロディの繰り返し構造や A メロ，サビなどの音楽的要素を考慮し，より自然な歌詞の生成を目指す．

謝辞　本研究は，RWC 研究用音楽データベースを利用した．本研究は科研費 JP16J05945 の助成を受けた．

参考文献

[1] Tatsuji Ueda. よくわかる作詞の教科書. YAMAHA music media corporation, 2010.
[2] Ananth Ramakrishnan A, Sankar Kuppan, and Sobha Lalitha Devi. Automatic generation of tamil lyrics for melodies. In *Proc. of the Workshop on Computational Approaches to Linguistic Creativity*, pages 40–46, 2009.
[3] Hugo R. Gonçalo Oliveira, F. Amilcar Cardoso, and Francisco C. Pereira. Tra-la-lyrics: an approach to generate text based on rhythm. In *Proc. of 4th International Joint Workshop on Computational Creativity*, pages 47–55, 2007.
[4] Gabriele Barbieri, François Pachet, Pierre Roy, and Mirko Degli Esposti. Markov constraints for generating lyrics with style. In *ECAI 2012*, pages 115–120, 2012.
[5] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech 2010*, pages 1045–1048, 2010.
[6] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *ISMIR 2002*, volume 2, pages 287–288, 2002.
[7] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.
[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML 2015*, volume 37, pages 448–456, 2015.