

リリック特徴を反映したキーワードに基づく ラップリリック生成

村上 和聴^{1,a)} 寺井 あすか^{1,b)}

概要: 本研究は、ラップバトルにおけるリリック生成システムの開発を目的とする。ラップバトルは即興でリリック（歌詞）を作成し、それを対話形式で披露しあう競技であり、技巧のあるライム（韻）を用いたリリックを創造する技術が求められる。ライムとは「明確」と「生活」のように、単語の母音列が同一の語のことである。このライミング（韻を踏むこと）の独特性から、近年ラップバトルなどのラップに関わるコンテンツが流行している。しかしながら、ラップの知識を持たない人がライミングを行うことは容易ではない。そこで本研究では、日本語の格関係を考慮したキーワードとラップリリックの特徴、テンプレートを用いた文生成手法を組み合わせた新しいラップリリック生成手法を提案した。さらに、実際に生成したリリックの日本語文およびラップリリックとしての妥当性に関する評価実験を実施した。

Rap lyric generation based on keywords reflecting lyric features

KAZUKI MURAKAMI^{1,a)} ASUKA TERAI^{1,b)}

1. はじめに

近年ラップという文化が流行となっており、その中でもラップバトルというジャンルが特に注目を集めている。ラップバトルに関するコンテンツが多く制作されている。例えば、地上波やネット番組など様々なメディアで放送されたラップバトル番組「フリースタイルダンジョン」、日本最大級のラップバトル「ULTIMATE MC BATTLE」などの多様な大会が開催されている。

ラップバトルとは即興で作成したラップリリックをお互いに披露しあい、どちらが優れているかを競う競技である。ラップバトルのイメージを図1として提示する。

本研究では、各人の発話の単位を「バース」(図1の吹き出し)、バース内の文の単位を「リリック」(図1の下線部)、各リリック内で韻を踏んでいる単語を「ライム」とし(図1の太字)、特にリリックに関し、バース内の1行目を「先行リリック」、2行目を「後続リリック」とする。

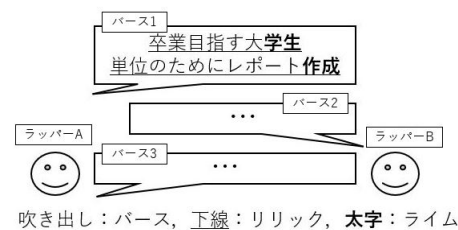


図1 ラップバトルのイメージ図

ライミング（韻を踏むこと）の知識のない人がラップリリックを即興で作成することは困難であり、コンテンツ制作に手軽に手を出しにくい。そこで、本研究ではラップバトルにおけるリリック自動生成として、先行リリックを入力することでの後続リリックの生成を行うシステムを提案する。このシステムによって、ライミングの知識のない人へのラップコンテンツの作成や創作支援が期待できる。また、ライムが考慮された歌詞はリズムカルで心地よい響きを持つ[1]ということが示されていることから、ラップだけでなく多様なジャンルの音楽での作詞支援も期待できる。

ラップリリック生成に関する先行研究として、深層学習を応用した研究も行われている。Wuら[2]はニューラル

¹ 公立はこだて未来大学
Future University Hakodate
^{a)} b1018190@fun.ac.jp
^{b)} aterai@fun.ac.jp

ネットワークを用いたラップリリック生成手法と記号的理論を用いたラップリリック生成手法の2種類の手法を提案し、それぞれの手法の比較を行った。Potashら[3]はLSTMを用いたラップリリックの生成に取り組み、特定のラッパーのスタイルに似ているが、既存のものとは異なるリリックの自動生成を行った。また、Xueら[4]が提案した生成モデルでは、Transformerを用いてリズムを考慮したライム生成を実現している。

このように、深層学習を用いた英語でのラップ生成に関する研究は多くなされている。一方、日本語でのリリック生成に関する研究はあまり行われていない。これは、ラップという音楽ジャンルが英語圏発祥でありデータ数が多いことに対し、日本語での研究はデータが少ないことに起因すると考えられる。ラップバトルという場面での応用を考慮した研究となると、さらに数が少なくなるが、深層学習を用いた日本語でのラップ文生成の研究に三林ら[5]の研究がある。

三林ら[5]は機械学習を用いたバース生成とテンプレートを用いたバース生成の2手法を提案・比較した。その結果、機械学習を用いた手法では、全体的に支離滅裂な内容となっており、日本語の文章としてうまく出力されなかった。対して、テンプレートを用いた手法では、文構造を維持したままバース生成を行うため、日本語文としての構造がある程度保たれた出力結果となった。この結果に対し、機械学習を用いたバース生成手法の問題点として、データ不足が挙げられていた。

また、佐藤ら[6]は、ラップリリックのパターン推定に基づき、入力された先行リリックに対し、ライムとなる単語を出力するシステムを提案した。ライムの形態素やライムとなる単語間の意味的距離にパターンがあると仮定し、実際のラップバトルにおけるリリックデータ分析からそれらのパターンを推定した。その結果、ライムの形態素として、名詞および名詞と助詞を組み合わせた語を用いてライミングが行われていることが示された。また、そのライムに用いられている名詞間の平均意味的距離に基づき、入力文(先行リリック)に含まれる名詞に対してライムとなる単語を出力するシステムを提案した。

本研究では、佐藤らの先行研究[6]を発展させ、入力された先行リリックに対しライムを生成し、先行リリックをテンプレートとして用いることで後続リリックを生成するシステムの生成を行う。また、バース内の連続するリリックが持つ特徴の抽出を行い、先行リリックをテンプレートとして用いることの妥当性を確認するとともに、抽出された特徴に基づく生成システムの構築を行う。

2. 利用した技術

ここでは本研究で用いる技術について簡潔な説明と、本研究との関係性を述べる。

2.1 MeCab

MeCab[7]は日本語形態素解析エンジンの一つであり、京都大学情報学研究所と日本電信電話株式会社コミュニケーション科学基礎研究所によって共同開発され、オープンソースとして公開されている。形態素とは、言語において意味を持つような最小の単位のことを指す。そして形態素解析とは、自然言語の文を形態素単位に分割し、各形態素に品詞などの情報を付与する処理のことを指す。

英語などの空白で文を区切るような言語では、空白で単語を分割することができるが、日本語の場合は品詞のような形態素ごとの情報を記述した辞書を参照して形態素解析を行う。この際、「テレビ東京」のように分割する位置が複数箇所考えられる場合がある。この問題に対処するため、日本語形態素解析では以下のアルゴリズムが用いられている。

- (1) 文に対して辞書を参照し、可能な形態素分割を全て行う。
- (2) 1で分割した形態素全てに対し、一定の評価基準に基づき最適な分割を選択する。

2の一定の評価基準とは、最小コスト法によって計算されている。MeCabでは、評価を計算する際のコスト値推定としてCRF(Conditional Random Fields)が採用されている。CRFの採用により、品詞、単語長、辞書の変更に對し柔軟にかつ低コストで対応でき、解析速度も維持できる。本研究ではこの技術をライムの提案や入力文解析に用いた。

2.2 Word2vec

Word2vec[8]は、単語の埋め込みを生成を行う技術である。Word2vecは複数のモデルから構築されており、これらのモデルは単語の意味的関係性を再構築するように訓練されたニューラルネットワークであり、大きなコーパスから一つのベクトル空間を生成する。CBoW(Continuous Bag-of-Words)モデルおよびskip-gramモデルという二つのモデル構造を使用し、単語の分散表現を生成する。CBoWモデルでは、文脈中の単語から対象単語が現れる条件付き確率を最大化する。つまり、前後の単語から対象単語を予測する。逆にskip-gramモデルでは、周辺単語予測の誤差を最小化する。つまり、対象単語から前後の単語を予測する。対象単語に近いほど、前後の単語の重みを大きくする。これらモデルにより、文章中の単語をベクトルに変換してその意味を数値化することができる。本研究ではこの技術をライム提案時の意味的距離の計算に用いた。

2.3 BERT

BERT(Bidirectional Encoder Representations from Transformers)[9]は自然言語処理モデルの一つである。BERTの学習は二つの段階があり、一つ目は事前学習、

二つ目はファインチューニングと呼ばれる学習である。事前学習では、学習が最適となるような重みの初期値を、事前に別のニューラルネットワークを学習して求める。また、BERTの特徴として、文を両方向(「左から右」および「右から左」)で学習するため、BERTでは2つの事前学習を行なう。それがMLM(Masked Language Modeling)とNSP(Next Sentence Prediction)である。MLMは、入力文の15%のトークンをマスクし、元のトークンを当てはめるタスク。つまり、穴埋め問題を解くようなタスクを行う。そしてNSPは、ある2文を選択し、それらが連続した文かどうかを判定するタスクを行う。そして、ファインチューニングでは、事前学習で求めた重みを初期値として、新たに追加したモデルを合わせた全体の微調整を行う。ファインチューニングを行い、モデル全体のデータを再学習させることで、汎化性能がより向上する。また、両方向性を持ったモデル(BERT)においてファインチューニングを行うことで、より精度を向上させた。

また、BERTを文章ベクトルのタスクに応用したSentence-BERT(S-BERT)ト[10]というものがある。Sentence-BERTはBERTをファインチューニングし、文章ベクトルを生成する手法である。ある文章とそれに類似する文章のペアを学習データとし、似た文章から生成される文章ベクトルが似たベクトルになるようにファインチューニングを行う。

本研究では、BERTをリリック生成時の単語補完に、S-BERTをリリックの選定時の文章ベクトルの計算に用いていた。

3. ラップリリック特徴抽出

先行研究では出力がライムであるため、元データのライム同士の関連のみを考慮した分析が行われたが、本研究では出力をリリックとするため、連続したラップリリックが持つ特徴を明らかにすることを目的とした分析を実施する必要がある。そこで、ラップリリックの特徴としてライムのパターン、ライムの位置、リリックの長さを調査項目として設定し、MeCabを用いた形態素解析を行い、リリック内の単語を抽出した上で分析を行った。

3.1 方法

3.1.1 データ概要

韻を踏む単語の検索が可能なWebサービス韻ノート[11]にまとめられている、実際のラップバトルで用いられた1530バースを対象とした。

3.1.2 ライムのパターン

ライムを構成する母音列が連続するリリック内で変化する可能性があるため、ライムのパターンの分類を行う。ライムのパターンを分類するために、実際のバトルで用いられたリリックのデータにタグ付けを行う。タグ付けの内容

卒業目指す大学生

ououe a u aiauei → 9

単位のためにレポート作成

aniaoa e i e oo o auei → 10

図2 ライムを除いたリリックの母音数

は、同一の母音列を含むライムに同一のタグをつけるというものである。

バース内のライムパターンとして、以下に示す5パターンを定義する。

- (1) AA'BB': 同一母音列から成るライムが隣接する
- (2) ABB'A': 同一母音列から成るライム同士が離れている
- (3) ABA'B': 同一母音列から成るライムが交互に現れる
- (4) AA'A": ライムの母音列が常に同一
- (5) その他: 上記のいずれにも該当しない

3.1.3 ライムの位置

ライムがリリック内のどの位置に配置されていることが多いのかを検証する。リリック内におけるライムの位置としては、「文頭」「文末」「文頭と文末」「その他」の4点を分類項目とする。「文頭と文末」というのは、初めは文頭にライムがあるが、リリックの途中で文末に位置が変化しているものである。

3.1.4 リリックの長さ

リリックがおおよそ何個の母音から構成されているのかを検証する。ライム自体はキーワードとして扱うということに加え、ライムに用いられる単語によってライムの母音数は変化するということから、ライム自体の母音数を分析の際に含むとノイズとなるため、リリック内のライムを除いた部分の母音数を対象としてこの検証を行う。リリック内のライムを除いた部分の母音数とは、図2の太字(ライム)部分を除いた部分の母音数のことである。

3.2 ライムの提案

先行研究[6]では、入力文に含まれる名詞と単語の母音列が同一である単語に対し独自で設定した意味的距離に近い順でライムを提案していた。しかし、入力単語との意味的距離が近い順とした場合と出力結果に差はみられなかったため、本研究では入力単語との意味的距離が近い五つのライムを提案する。また、先行研究では、入力文に含まれている単語がライムとして提案される単語と重複することを認めていたが、本研究では出力されるリリックに幅を持たせるために重複しないようにした。

3.3 結果

3.4 ライムのパターン

パターンを分類した結果、パターン1(AA'BB')が114個、パターン2(ABB'A')が2個、パターン3(ABA'B')が4個、パターン4(AA'A")が1398個、その他が12個とい

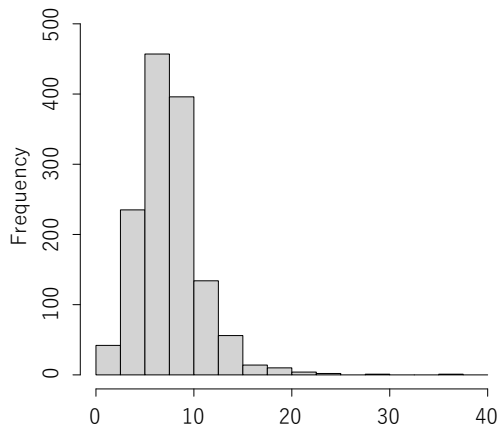


図3 母音数の平均

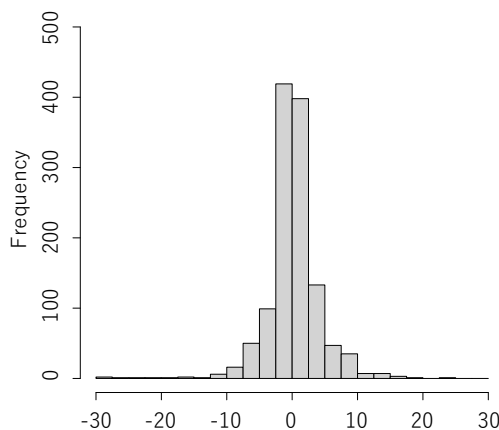


図4 母音数間の差の平均値

う結果となった。

91.4%のバースにおけるライムはパターン4(AA'A":ライムの母音列が常に同一のもの)に分類された。そこで、以下の分析ではパターン4のバースのみを対象として、分析を行なった。

3.5 ライムの位置

ライムの位置を分類した結果、文頭が58個、文末が942個、文頭と文末が245個、その他が153個という結果となった。リリックの文末にライムが配置されているものが1187個あり、84.9%を占めていることが分かった。

3.6 リリックの長さ

バース内のリリックの母音数の平均値のグラフを示す(図3)。リリック内の平均母音数の64.2%が5より大きく10以下の範囲内にあるということが分かった。

次に、バース内の連続するリリックの母音数の差の平均値のグラフを示す(図4)。差の平均値の70.3%が-2.5より大きく2.5以下の範囲にあるということが分かった。

3.7 考察

以上の分析より、ラップリリックの特徴として、次の3点が挙げられる。

- (1) ライムの母音列が常に同一であることが多い
- (2) (ライムの母音列が同一の場合) ライムはリリックの文末に配置されることが多い
- (3) (ライムの母音列が同一の場合) 連続するリリック間の母音数の差は ± 2.5 の範囲内にあることが多い

4. リリック生成システム

抽出されたラップリリックの特徴に基づき、入力された先行リリックに対して後続リリックを生成するシステムを提案する。具体的には、ライムの母音列が常に同一である・ライムはリリックの文末に配置されることが多い、という特徴に基づき、先行リリックの文末の単語のみをライムの対象とする。また、連続したリリックはライムの位置・母音数について類似した特徴を持つことから、先行リリックをテンプレートとした後続リリック生成を行う。

4.1 システムの概要

4.1.1 ライムの提案

先行研究[6]のライム提案システムに基づき、ライムの提案を行った。先行システムにおけるパターン推定により設定された基準に基づくライム提案方法と、先行リリックにおける入力単語との意味的距離に近い順によるライムの提案方法において顕著な差が見られなかった。そこで、本研究では、入力された先行リリックの文末の単語(名詞)を対象として、学習済みWord2vec[12]により推定した意味的距離の近いライム5個を取得することとした。

4.1.2 キーワード抽出

京都大学格フレームコーパス[13]を用いて、提案されたライムに関するキーワードの抽出を行う。格フレームとは、用言とそれに関する名詞を用言の用法ごとに整理したものであり、このコーパスを用いることで、文内でもともに用いられることの多い単語を抽出できる。例えば「生活」という名詞で検索すると、「生活」という単語とともに用いられることの多い用言「送る」「する」「営む」といった用言を頻度順に取得することができる。この単語群を抽出し、文生成のためのキーワードとすることによって、人間が作成したものに近い文が生成できると考えられる。

頻度順に取得したキーワードすべてをリリック生成のキーワードとして用いると膨大な時間と計算資源を要するため、頻度順の上位5個をキーワードとして以下のリリック生成を行う。ライムに対するキーワードが5個に満たない場合は、そのまま5個に満たないキーワードを用いることとした。また、提案するライムとして出力された単語が格フレームコーパスに含まれていなかった場合は、キーワード抽出を行わない。

4.1.3 リリック生成

リリック生成はテンプレートを用いた手法で行う。連続したリリックはライムの位置・母音数について類似した特徴を持つことから、入力した先行リリックをテンプレートとして後続リリックを出力する。まず、キーワード抽出で取得したキーワードと提案するライムを用い「キーワード+ライム(文末)」という形式のリリックを生成する。しかしながら、キーワードとライムだけでは入力された先行リリックに類似した母音数となるリリックを得られない可能性があるため、不足した部分を補うためにBERTを用いて、「*+キーワード+ライム(文末)」という形式のリリックにおける「*」部分に関する穴埋め処理を行う。穴埋め処理は、とBERTの日本語事前学習済みモデル[14]を用い、BERTの単語予測上位十個を対象とした。

さらに、取得された5種類のライムに対し、キーワードが各5個が抽出され、ライムとキーワード1組に対してBERTで予測される穴埋め上位10個、合計250個のリリックが生成される。これらのリリックから、助詞や記号から始まるような日本語文として不適なリリックを除去する。そして、S-BERTを用いて、残ったリリック・入力された先行リリックを文ベクトルに変換し、各ライムに関し、先行リリックとの意味的距離が最も近いリリックを後続リリックとして出力する。

4.2 出力結果

以下、入力文を「花火のような運命」とした出力結果を記述する。

4.2.1 ライムの提案

提案ライムとして、以下の5つの単語が出力された(上から順に意味的距離が近い順)。

- (1) 文明
- (2) 運勢
- (3) 君命
- (4) 軍勢
- (5) 軍籍

4.2.2 キーワード抽出

提案ライムとして出力された5つの単語に対して、それぞれ出力されたキーワードを示す。

- (1) 築く, 持つ, 持つ, 発達する, 築く
- (2) 一覧, 占う, 見る
- (3) 【キーワードなし】
- (4) 戦う, 率いる, 差し向ける, 現れる
- (5) 離れる, 得る, 置く, 剥奪する, 置く

4.2.3 リリック生成

後続ライムとして、提案ライムと抽出されたキーワードを用いて生成され、各ライムに関し、入力された先行リリックとの意味的距離が最も近いと推定されたリリックを示す。

- (1) 自然発達する文明
- (2) 通称占う運勢
- (3) 【出力結果なし】
- (4) その後現れる軍勢
- (5) 遠く離れる軍籍

5. 評価実験

5.1 方法

実際のラップバトルで用いられたバースを用い、先行リリックを提案システムに入力した際に出力されるリリックと、実際に後続するリリック(以下「オリジナルリリック」)を混合させた評価実験をWeb調査により実施した。この実験で用いたバースは、提案システムの対象となる、リリックの文末に名詞のライムを持つバースに関し、システムが少なくとも3種類のリリックを出力可能なもの、10種類を無作為に抽出した。

参加者には下記の2点を条件として、歌詞A(先行リリック)を提示した上で、各リリック(歌詞B)を5段階で評価してもらった。

- ラップの歌詞として適しているか
- 「歌詞A 歌詞B」という形式で歌詞Aに後続する歌詞Bとして適しているか

先行リリックに対する提案システムの出力リリック(最大5種)と、実際の後続リリックを1セットとして、全10セットについて評定してもらった。また、参加者数は10代から60代の男女139人だった。なお、参加者には実際のラップバトルで用いられたリリックが含まれていることは伝えなかった。

5.2 結果

システムの利用方法として、複数の出力リリックからユーザが良いと思ったものを1つ選択することを想定している。そのため、各セット内における、各参加者の出力リリックに対する評価点の最大値を提案システム生成リリックの評価点とみなした。そして、提案システムで生成したリリックとオリジナルリリックに対する評価を比較するため、提案システム生成リリックの評価点とオリジナルリリックの評価点に対し、各セットごとに対応のある t 検定を行った。その結果を表1に示す。なお表中では、提案システム生成リリックの平均評価点を $M1$ 、オリジナルリリックの平均評価点を $M2$ 、入力リリックの母音数を v 、提案システムで出力されたリリック数を l とし、有意性はbonferroni補正を適用した。

表1より、セット2($t(138) = 4.027, p < .001$)、セット6($t(138) = 8.806, p < .001$)、セット8($t(138) = 10.071, p < .001$)、セット9($t(138) = 13.021, p < .001$)、セット10($t(138) = 6.778, p < .001$)にて提案システム生成リリックが有意に評価が高く、セット3($t(138) = 5.103, p < .001$)、

表 1 評価結果の表

| セット | M1 | M2 | t 値 | p 値 | 有意性 | v | l |
|-----|-------|-------|--------|------------|-----|----|---|
| 1 | 3.367 | 3.417 | 0.401 | 0.689 | n.s | 14 | 4 |
| 2 | 3.942 | 3.475 | 4.027 | $p < .001$ | ** | 6 | 5 |
| 3 | 3.194 | 3.835 | 5.103 | $p < .001$ | ** | 7 | 3 |
| 4 | 3.655 | 3.619 | 0.265 | 0.791 | n.s | 13 | 5 |
| 5 | 3.827 | 4.583 | 7.849 | $p < .001$ | ** | 8 | 4 |
| 6 | 3.928 | 2.835 | 8.806 | $p < .001$ | ** | 8 | 5 |
| 7 | 4.108 | 4.079 | 0.309 | 0.757 | n.s | 14 | 5 |
| 8 | 4.086 | 2.892 | 10.071 | $p < .001$ | ** | 11 | 5 |
| 9 | 4.245 | 2.727 | 13.021 | $p < .001$ | ** | 10 | 5 |
| 10 | 3.993 | 3.295 | 6.778 | $p < .001$ | ** | 8 | 5 |

** $p < .01$

セット 5($t(138) = 7.849, p < .001$) でオリジナルリリックの評価が有意に高いという結果になった。対して、セット 1($t(138) = 0.401, n.s$)、セット 4($t(138) = 0.265, n.s$)、セット 7($t(138) = 0.309, n.s$) では有意差はみられなかった。また、同様に各参加者の 10 セットに関する提案システム生成リリックの平均評価点、オリジナルリリックの平均評価点を用いて、対応のある t 検定を行った結果、0.1% 水準で有意差がみられ ($t(138) = 8.17, p < .001$)、提案システム生成リリックの平均評価点 (3.83) は有意にオリジナルの平均評価点 (3.48) より高いという結果が得られた。

5.3 考察

評価実験の結果より、提案システム生成リリックの平均評価点 M1、オリジナルリリックの平均評価点 M2、入力リリックの母音数 v 、提案システムで出力されたリリック数 l を対象として考察する。

まず、入力リリックの母音数でセットごとに分けると、 t 検定で平均評価点の間に有意差が見られなかったセット 1, 4, 7 の入力リリックの母音数は、それぞれ 14, 13, 14 とセット中でも多い傾向があった。一方、平均評価点の間に有意差がみられたセットの入力リリックの母音数は、6 から 11 と評価点に有意差がみられなかった場合よりも少ない傾向があった。なお、提案システム生成リリックの評価点が有意に高いという結果が得られた入力リリックの母音数は 6 から 11、オリジナルリリックの評価点が有意に高いという結果が得られた入力リリックの母音数は 7 と 8 となり、特に傾向はみられなかった。先行リリックの母音数が少ないほど、提案システムはオリジナルリリックとの評価に大きな差のある後続リリックを生成していた。先行リリックの母音数が多い場合、提案システムではオリジナルの評価点を超えるリリックを生成できていなかったといえる。この点に関して、BERT の穴埋めを繰り返し行う過程で、適切な穴埋め単語を見つけられず、母音数の多いリリック作成の精度が低下した可能性が考えられる。

次に、平均評価点から考察すると、10 セット中 6 セットは提案システム生成リリックがオリジナルリリックより高く評価され、そのうちの 5 セットにおいて有意であった。10 セット中 4 セットはオリジナルリリックが提案システム生成リリックより高く評価され、そのうちの 2 セットにおいて有意であった。このことから、提案システムは、実際にラップバトルで用いられたリリックと比べても、遜色のないリリックを含む出力ができるといえる。

最後に、提案システムで出力されたリリック数 l に関し、システムの構造上 l の最大値は 5 であり、実験では最小値 3 となるようなリリックを入力リリックとして用いた。 l が 3 または 4 だったセットは 1, 3, 5 であり、どのセットでも提案システム生成リリックの平均評価点はオリジナルリリックの平均評価点より低くなっている。対して、それ以外の場合は l が 5 となっており、提案システム生成リリックの平均評価点はオリジナルリリックの平均評価点より高くなっている。 l の値によって評価が分かれていることから、システムで最大数までリリックを出力できるようなリリックを入力とした場合、選択肢が増加し、高い評価が得られるリリックを生成できる確率が増加していたと考えられる。

6. 結論

本研究では、バース内の連続するリリックが持つ特徴に基づき、入力された先行リリックに対しライムを生成し、格フレームコーパスを用いたキーワード抽出・先行リリックをテンプレートとして用いた文生成を行うことで、後続リリックを生成するシステムを提案した。さらに、実際のラップリリックと生成されたリリックを用いた評価実験により、実際のラップリリックと同等以上のラップリリック生成が可能であることが示唆された。

今回提案した手法では、各リリック内での文脈や、先行リリックと後続リリックのライム同士の意味的距離は考慮していたが、先行リリックと後続リリックの文脈的つなが

りは考慮されていない。本来のリリックでは、先行リリックと後続リリックが文脈的につながっていることが予想されるため、これを考慮することでよりラップの特徴をとらえたりリリックを生成できると考えられる。また、出力されるリリックの個数が入力リリックに左右されてしまうととも、生成されたりリリックの評価に影響していた。入力リリックにかかわらず、出力されるリリックの個数を安定して最大数まで生成できるようにすることで、より評価の高いリリックが生成される可能性が増加すると考えられる。

謝辞 本研究は JSPS 科研費 (19K12121, 16H02835) の助成を受けたものである。

参考文献

- [1] 鎌田里沙子, 山西良典, 西原陽子, 福本淳一, “表現特徴に着目した歌詞の印象的フレーズ抽出”, 情報処理学会研究報告, Vol. 2013-MUS-101, No. 6, pp. 1-6, 2013.
- [2] Dekai Wu and Kartek Addanki. Neural Versus Symbolic Rap Battle Bots. In *Proceedings of ICMC' 15*, pp. 210-213, 2015.
- [3] Peter Potash, Alexey Romanov, and Anna Rumshisky. Ghostwriter: Using an LSTM for automatic rap lyric generation. In *Proc. of EMNLP' 15*, pp. 1919 - 1924, 2015.
- [4] Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, Tie-Yan Liu. DeepRapper. Neural Rap Generation with Rhyme and Rhythm Modeling. In *Proceedings of ACL-IJCNLP'21*, pp. 69-81, 2021.
- [5] 三林 亮太, 橋口 友哉, 山本 岳洋, 大島 裕明, “ラップバトルにおけるライムの意味類似性を考慮したバース生成システム”, DEIM 2021, 2021.
- [6] 佐藤 颯, 寺井あすか, “パターン推定に基づくラップリリックの生成”, 情報処理北海道シンポジウム, 2020.
- [7] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto. Applying Conditional Random Fields to Japanese Morphological Analysis, *Proc. of the 2004 Conf. on Empirical Methods in Natural Language Processing (EMNLP-2004)*. pp.230-237. 2004.
MeCab <https://taku910.github.io/mecab/>
- [8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality, *Advances in Neural Information Processing Systems 2013*, pp. 3111-3119. 2013.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. s.l. : arXiv preprint arXiv:1810.04805. 2018.
- [10] Nils Reimers, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. s.l. : arXiv preprint arXiv:1908.10084. 2019.
- [11] 韻ノート <https://in-note.com/>
- [12] Word2Vec http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/
- [13] 林部祐太, 河原大輔, 黒橋禎夫 “格パターンの多様性に頑健な日本語格フレーム構築”, 情報処理学会, 自然言語処理研究会, 224-14, pp. 1-8, 2015.
- [14] 訓練済み BERT モデル
<https://www.nlp.ecei.tohoku.ac.jp/news-release/3284/>