

# ラップバトルにおける逆向き生成による ライムを含む返答バース生成

三林 亮太<sup>1,a)</sup> 山本 岳洋<sup>1,b)</sup> 佃 洗撰<sup>2,c)</sup> 渡邊 研斗<sup>2,d)</sup>  
中野 倫靖<sup>2,e)</sup> 後藤 真孝<sup>2,f)</sup> 大島 裕明<sup>1,g)</sup>

受付日 2023年9月10日, 採録日 2023年12月25日

**概要:** ラップバトルとは、2名のラッパーが即興のラップを交互に行う競技である。ラッパーが1ターン内で行う即興のラップをバースと呼び、バースは複数のラップ文で構成される。本論文では、相手のバースに対するライム（韻）とアンサーを同時に考慮した返答のバース生成手法を提案する。提案手法では、生成モデルである BERT2BERT を用いてラップ文を生成し、それらの文を BERT を用いて適切に並べ替えることでバースを構成する。ラップ文を生成する際には、ライムとなる語を文末に含めることが重要である。しかし、文頭から順方向に文を生成する従来の文生成モデルでは、そのような語を文末に含めることが困難である。それに対して提案手法では、文末から逆方向に文を生成することで、文末にライムを含むラップ文を高い確率で生成することを実現した。ラップ文生成モデルを学習するために、我々は 6,791 バースからなる独自のラップバトルコーパスを作成した。そのうちの 100 件のバースに対して、提案手法および順方向での文生成手法により、返答バースを生成して人手で評価を行い、バースの自然さ、ライムの質、アンサーの質のいずれも提案手法の方が優れていることを示した。

**キーワード:** ラップバトル, バース生成, BERT2BERT, ライム, アンサー

## Response Verse Generation from Reverse Direction Including Rhymes in Rap Battles

RYOTA MIBAYASHI<sup>1,a)</sup> TAKEHIRO YAMAMOTO<sup>1,b)</sup> KOSETSU TSUKUDA<sup>2,c)</sup> KENTO WATANABE<sup>2,d)</sup>  
TOMOYASU NAKANO<sup>2,e)</sup> MASATAKA GOTO<sup>2,f)</sup> HIROAKI OHSHIMA<sup>1,g)</sup>

Received: September 10, 2023, Accepted: December 25, 2023

**Abstract:** Rap battle is a competition in which two rappers improvise rap verses alternately. The rap that is delivered in one turn is called a verse, and a single verse is generally composed of several rap sentences. In this paper, we propose a verse generation method that simultaneously considers rhyme and answer in response to the opponent's verse. Our approach uses a language generation model BERT2BERT to generate rap sentences and constructs a verse by appropriately arranging them using a BERT model. When generating rap sentences, it is important to include words that rhyme with a specific word in the opponent's verse, but it is difficult to include such words using a conventional sentence generation model that generates sentences in a forward direction from the beginning of the sentence. To address this issue, our proposed method trains the model to generate sentences in a reverse direction from the end of the sentence, which enables the model to generate rap sentences that highly likely have rhymes at the end. To train the model, we constructed our own rap battle corpus consisting of 6,791 verses. We generated verses for 100 input verses using both the proposed method and the conventional forward sentence generation method and evaluated them manually. The results demonstrated that the proposed method outperformed the conventional method in terms of the naturalness of verse, the quality of rhyme, and the quality of answer.

**Keywords:** rap battle, verse generation, BERT2BERT, rhyme, answer

## 1. はじめに

ラップバトルとは、2名のラッパーが即興のラップを交互に行う競技である。ラッパーが1ターン内で行う即興のラップをバースと呼び、バースは複数のラップ文で構成される。図1にラップバトルの例を示す。ラッパーAの「まずはリズムに乗せる小節/俺のライムは特別」というバースに対して、ラッパーBは「一体お前のどこが特別/ラップ上手い順ならお前は後列」というバースを返答する。このようなやりとりを複数回繰り返すことでラップバトルは終了する。ラップバトルの勝敗は、観客や審査員による多数決の評価によって決められる。ラップバトルでは、バースの質が勝敗の判断基準となる。

バースの質の判定には、一般的にフロー、ライム、アンサーの3つの要素が重要な役割を担う[1]。フローとは、ラッパーによるラップのリズムやアクセントなどの音声レベルの表現を指す。一方、ライムとアンサーは、以下のよう

- ライムとは、同じ母音を持つ単語のペアを指す。たとえば、図1のバース1の「特別」とバース2の「後列」のペアは、どちらも同じ母音列“*oueu*”を持っているため、ライムのペアと見なされる。ライムはバース内の任意の位置に含むことができるが、バース内のそれぞれの文末に含めることが一般的である。
- アンサーとは、相手のバースに関連する返答のことを指す。たとえば、図1のバース2の「一体お前のどこが特別」は、バース1の「俺のライムは特別」と意味的に関連しているため、アンサーが含まれていると見なされる。

質の高いバースでは、ライムとアンサーの両方が含まれており、それがフローで表現されている。さらに、それらライム、アンサー、フローの質の高さも勝敗の判断に影響する。しかし、そのようなバースを即興で考える必要があるラップバトルでは、質の高いバースをつねに返答することは熟達者であっても難しい課題である。

ラップバトルの大会は多くの国で開催されている。たとえば、アメリカでは「True Freestyle Rap Battle」、韓国では「UNPRETTY RAPSTAR」というラップバトルの

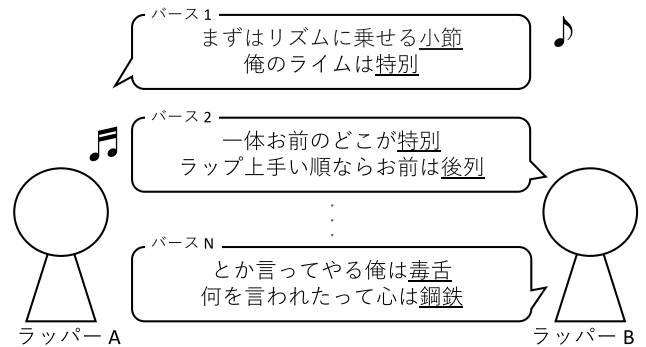


図1 ラップバトルの例

Fig. 1 An example of a rap battle.

テレビ番組がある。日本では、「凱旋 MC battle」や「戦極 MCBATTLE」など、人気のラップバトル大会がある。ラップバトルには、これらのようなプロのラッパーが参加する大会だけでなく、アマチュアのラッパーが参加する大会も多い。前述したとおり、ラップバトルでは相手のバースに対して、いかにライムを含んでアンサーをするかが重要である。しかし、即興でそのような質の高いバースを思いつくのは、初心者には難しい。もし、相手のどのようなバースに対しても、自動的に質の高いバースを生成してくれるシステムがあれば、生成されたバースを参考にラップバトルの練習をすることができるようになる。

本論文では、そのようなシステムの実現に向けた第1歩として、ラップバトルにおけるライムとアンサーを考慮した返答バースを生成する手法を提案する。なお、本手法では、テキストレベルの表現であるライムとアンサーに着目し、音声レベルの表現であるフローは扱わない。

提案手法では、生成モデルである BERT2BERT [2] を用いて複数のラップ文を生成し、それらを BERT を用いて順番に並べる、という2つのステップで返答バースを生成する。つまり、ラップ文の生成とラップ文の並べ替えの2つのステップでバースを構築する。1つ目のステップでは、相手のバース内の文を入力し、ライムを考慮したラップ文を生成する。通常、文生成モデルは文頭から順方向に文を生成する。そのため、文末にライムを生成する場合、それまでの文脈に沿った単語を生成する必要があり、ライムにできる単語の候補が制限されるという問題がある。詳細については、2章で後述する。それに対して、本手法では文末から逆方向にラップ文を生成することで、文末にライムを容易に含めることができるようにする。特定のライムを満たす単語は多数存在するが、本手法では意味類似性を考慮したライムも考慮できるように、相手のバースから適切な単語を検索する。2つ目のステップでは、生成された複数のラップ文に基づいて、それらの連結確率を考慮する BERT [3] を学習し、それを用いて順番に並べて適切なバースを構成する。

本研究の貢献を以下に示す。

<sup>1</sup> 兵庫県立大学情報科学研究科  
Graduate School of Information Science, University of Hyogo, Kobe, Hyogo 651-2197, Japan

<sup>2</sup> 産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan

a) af22h007@guh.u-hyogo.ac.jp

b) t.yamamoto@gsis.u-hyogo.ac.jp

c) k.tsukuda@aist.go.jp

d) kento.watanabe@aist.go.jp

e) t.nakano@aist.go.jp

f) m.goto@aist.go.jp

g) ohshima@ai.u-hyogo.ac.jp

- ラップバトルにおいて、ライムとアンサーを考慮した返答バースの生成手法を提案した。
- 提案手法のモデルを学習するために、6,791 バースからなるラップバトルのコーパスを作成した。
- 評価実験を行い、逆方向にラップ文を生成する提案手法の方が、バースの自然さ、ライムの質、アンサーの質のいずれにおいても、順方向にラップ文を生成する手法を上回ることを示した。

本論文は、国際会議に採択された我々の英語論文 [4] に評価実験を追加し、日本語で記述したものである。

## 2. 関連研究

### 2.1 テキスト生成

近年のテキスト生成手法は、ディープラーニングを用いた手法が一般的である。たとえば、初期の段階では、RNN [5] や LSTM [6] などの自己回帰構造を用いた seq2seq [7] モデルが採用されていた。seq2seq モデルは、テキスト情報を集約する Encoder と、集約された情報からテキストを生成する Decoder を持ち、Encoder-Decoder モデルとも呼ばれる。その後、テキスト情報を効果的に選択する Attention と呼ばれる機構が提案された。さらに最近では、Attention を利用した Transformer [8] が提案されている。

BERT [3] と GPT-2 [9] は Transformer の構造を利用した代表的なモデルである。BERT [3] はテキストの特徴抽出器であり、Transformer を多層化した Encoder のみのモデルである。テキストの情報を双方向から考慮するため、テキストの文脈を考慮することができる。GPT-2 [9] は、テキスト生成タスクでよく使われる Decoder のみのモデルである。BERT とは異なり、GPT-2 は単方向のテキスト情報のみを考慮する。これらのモデルは一般的に、大規模なコーパスで事前学習して、汎用的な言語知識を獲得した後に使用される。事前学習モデルを特定のタスク用にファインチューニングすることで、そのタスクに特化したモデルを学習することができる。

BERT2BERT [2] は、BERT の事前学習されたパラメータを Transformer の Encoder と Decoder の初期パラメータとする生成モデルである。BERT [3] で得られた事前学習重みを利用することが、生成タスクにおいて有効であることが示されていることから、本研究では BERT2BERT をラップ文生成に利用する。

### 2.2 指定語句を含む文生成

Yamada ら [10], [11] は、指定語句を中心に文を双方向に生成し、双方向に生成した文を結合することで、指定語句を含む文を生成する手法を提案している。我々の研究では、ラップバトルにおいて基本的な技法である、文末にライムを含むことを目標としているため、双方向ではなく、文末のライムから逆方向への生成のみを行うことで目標を

達成する手法に取り組む。

### 2.3 ラップ生成

ラップの歌詞生成を目的とした研究では、生成された歌詞内の単語を置き換えることでライムを考慮する方法 [12] や、ライムを含む歌詞データを学習して、ライムを含んだ歌詞を生成する方法が提案されている [13], [14], [15], [16]。

Rapformer [12] では、Transformer [8] を用いてラップ文を生成し、生成された文末の単語を、母音が似ている他の単語に置き換えることでライムを考慮している。しかし、この置換方法では、置換する文末の単語はすでに生成された文の文脈に合った単語でなければならず、選択する単語が制限される。

歌詞内のライムを学習するラップ生成手法にも同じく制限がある [13], [14], [15]。Potash ら [13] は LSTM モデルを使って既存のラップ歌詞データから新しいラップ歌詞を生成する GhostWriter という手法を提案した。Malmi ら [14] が提案した DopeLearning は、生成されたラップ文と直前のラップ文の類似度を比較することで、最適なラップ文を選択する手法である。Manjavacas ら [15] も同様に LSTM を用いてラップの歌詞を生成する手法を提案している。これらの手法では、歌詞データを基にライムを文中に生成するようにモデルを学習している。モデルは、文頭の単語から文末の単語まで順方向に単語を生成し、ライムはいずれかの位置に生成される。しかし、生成するライムは、前の単語と文脈に合った単語でなければならず、生成する単語の候補は制限される。また、これらの手法はライムが生成される割合を向上させる制約を考慮しているが、ライムを必ず生成するという制約は考慮していない。

一方、Xue ら [16] は、Transformer-Decoder を用いて拍を考慮しながらラップを生成する DeepRapper という手法を提案している。DeepRapper はバースを文末から逆方向に生成するモデルを学習している点では、本研究の提案手法と似ている。しかし、DeepRapper を始めとした多くの既存手法はバースのみを生成する Transformer-Decoder を学習しているため、ラップバトルに対応することができない。それに対して我々の提案手法は、ラップバトルを前提に入力文からアンサーを生成する Transformer-Encoder-Decoder を学習している点が異なる。また、DeepRapper では複数文で構成されるバースを一度に生成するのに対して、我々の提案手法は入力文に対してライムを考慮したラップを 1 文ずつ独立に生成した後に、生成された複数文を並べ替えることで 1 つのバースを構成する。学習データが限られている場合、バース数よりもそれを構成する文数の方が多いため、バース単位より文単位の方が学習をしやすい。

ラップバトルに焦点を当てた、ラップの生成に関する研究はいくつかある。Wu ら [17] は RNN の一種である、TRAAM (Transduction Recursive Auto-Associative

表 1 提案手法と関連研究の違い

Table 1 Differences between the proposed method and related research.

	ライムを含むか	ラップバトルに向けたバース生成であるか	ラップバトルのデータを使用しているか
Rapformer [12]	✓	-	-
GhostWriter [13]	✓	-	-
DopeLearning [14]	✓	-	-
Manjavacas et al. [15]	✓	-	-
DeepRapper [16]	✓	-	-
Wu et al. [17]	✓	✓	-
Shimon [18]	✓	✓	-
Ours	✓	✓	✓

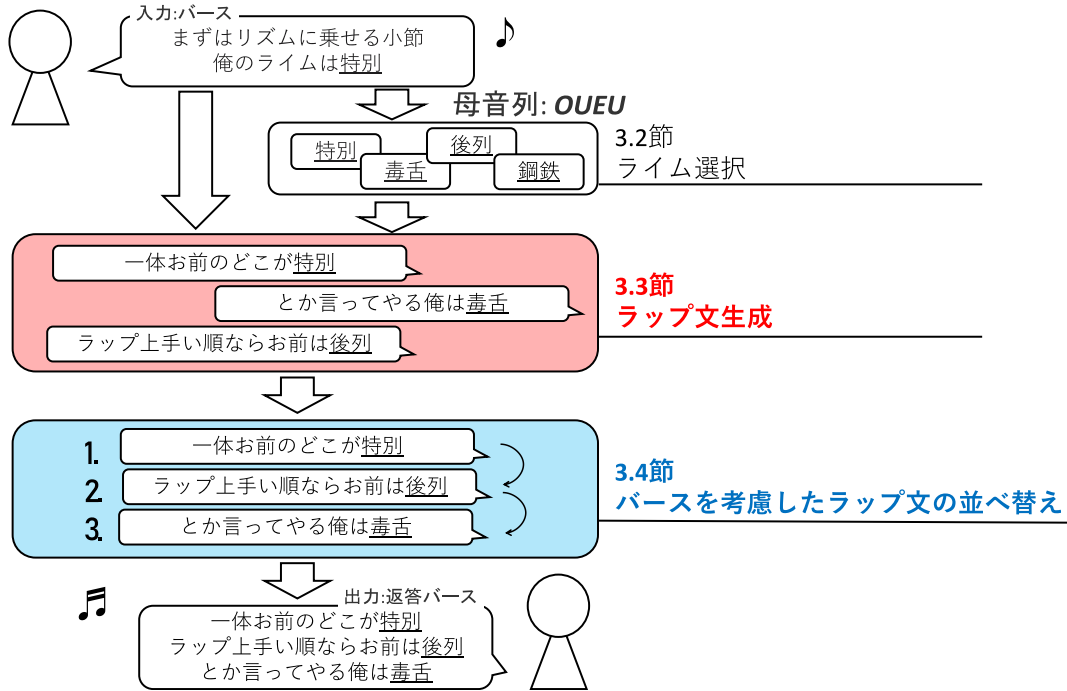


図 2 提案手法の全体図

Fig. 2 Overview of the proposed method.

Memory) と呼ばれるモデル構造を用いて、ラップバトル用のチャットボットシステムを開発した。このシステムは与えられたラップ文一文に対して、返答となるラップ文一文を生成することができる。Saveryら [18] によって提案された Shimon は、既存のラップ歌詞データを LSTM で学習して、ラップ文を生成する。さらに、生成されたラップ文を、ラップ文内のライムを基に並べ替えて歌詞を構築する。これらの研究では、独自のヒップホップ歌詞コーパスに基づく学習と生成が対象であり、ラップバトルのコーパスからモデルを学習することを対象としてない。つまり、仮にラップバトルに関するコーパスがあっても、それを活用できる手法ではない。ラップバトルは対話の競技であるため、相手の発言に対する返答の対話関係や、歌詞には出現しにくい、話し言葉の表現などをコーパスから学習する必要がある。

表 1 は上で紹介した研究をまとめたものである。我々の

研究は、文末にライムである任意の単語から逆方向にラップ文を生成するため、任意のライムを含んだ返答バースを生成できる点が特徴である。また、本研究では、日本語ラップバトルに特化したコーパスを独自に作成し、ラップ文生成モデルの学習に用いている。

### 3. ラップバトルにおける返答バース生成手法

ラップバトルにおける返答バース生成手法の概要を図 2 に示す。本手法では、一般的にラッパーが返答バースを生成する以下のプロセスを参考にする。まず、相手のバースからライムの対象となる単語を決める。その際、相手のバースに対するアンサーとなる単語を選ぶことが重要である。そして、そのような単語に対してライムとなる単語を含むバースを生成する。

以上のプロセスにならない、本手法では、まず相手のバース内のライムの対象となる単語に対して、ライムの候補

を見つける。そして、その中から意味的に関連するライムの集合を検索する (3.2 節)。次に、得られた各ライムを用いてラップ文を生成し、ラップ文の集合を得る (3.3 節)。最後に、ラップのバースを構成するために、生成されたラップ文を適切に並べ替える [14]。ラップ文の並べ替えでは、BERT を用いた隣接文予測により、バースを構築する (3.4 節)。

### 3.1 ラップバトルに関するコーパス作成

#### 3.1.1 ラップバトルコーパス

本研究では、ラップバトルにおける返答バース生成のために、独自のラップバトルコーパスを自作した。ラップバトルコーパスは、クラウドソーシングサービス Lancers<sup>\*1</sup> を利用して、日本語のラップバトル動画を文字起こしして作成した。文字起こしの対象とするラップバトルの動画は、チャンネル登録者数が 10 万人以上の YouTube のチャンネルである以下の 3 つを選択した。

- UMB<sup>\*2</sup>
- 凱旋 MC battle<sup>\*3</sup>
- 戦極 MCBATTLE<sup>\*4</sup>

Lancers で採用したワーカーは日本語を母国語とする者を対象とし、ラップバトルに精通しているかは問わなかった。ワーカーは、我々が作成した文字起こし作業用のウェブページにアクセスし、指定されたラップバトル動画の文字起こしを行った。具体的には、ラップ動画を別ウィンドウで再生しながら、そのラップ文を作業用ウェブページに用意されたテキストボックスに入力する。文字起こしの対象は動画内にあるラップバトルのすべてのバースである。広告などのラップバトル以外の部分は文字起こしの対象外とした。合計 194 人のワーカーによって 691 件のラップバトルを文字起こしした。文字起こし結果には質が低いものも部分的に含まれていたが、それについては著者が修正した。

また、コーパスのサイズを拡大するために、以下の番組で放送されたラップバトル 596 件についても、大学生 1 名が文字起こしを行った。

- BAZOOKA!!! 高校生 RAP 選手権
- フリースタイルダンジョン

コーパスの統計は表 2 のとおりである。ラップバトルの総数は 1,287 件で、それらに含まれるバースの総数は 6,791 件であった。ここでトークンとは、東北大学乾研が提供する BERT モデルのトークナイザ<sup>\*5</sup>で分割したサブワードを指す。動画内のラップはすべて日本語である。

<sup>\*1</sup> <https://www.lancers.jp/>

<sup>\*2</sup> <https://www.youtube.com/user/umbofficial>

<sup>\*3</sup> <https://www.youtube.com/channel/UCe-EvY8GrvYgx8PbwRBc75g>

<sup>\*4</sup> <https://www.youtube.com/user/senritumc>

<sup>\*5</sup> <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

表 2 ラップバトルコーパスの統計量  
Table 2 Statistics of the rap battle corpus.

ラップバトルの総数	1,287
バースの総数	6,791
バース内の文の総数	52,422
1 バースあたりの文の平均	7.71
1 文あたりのトークン数の平均	15.40

#### 3.1.2 ラップ文ペアデータ

ラップ文生成モデルを学習するには、ラップ文のペアが必要である。そこで、1 つのバースの中の連続する 2 つの文を取り出し、ラップ文のペアを作成した。1 つのバースが 3 つの文で構成されている場合、2 つのラップ文のペアが作成される。つまり、1 つ目と 2 つ目の文のペア、2 つ目と 3 つ目の文のペアが作成される。コーパスに含まれる 6,791 のバースから、合計 22,125 のラップ文のペアを作成した。今後、このペアデータを「ラップ文ペアデータ」と呼ぶ。

### 3.2 ライム選択

提案手法では相手のバースを基に、ライムとして使用する単語を選択する。このプロセスは以下の 3 つのステップからなる。

まず、相手のバースからライムの対象となる単語を抽出する。ラップバトルでラッパーがアンサーを考慮する方法の一種として、相手のバース内の単語をそのまま使用するテクニックがある。相手のバースは事前に行きたくないため、相手のバース内の単語をライムとしてバースに含んだ場合、即興で相手への発言に対して返答したと見なせる。さらに、相手のバース内の最後の単語をライムとして使用する場合、より高い即興性が要求されるため、高く評価される。そこで本研究では、相手のバースを形態素に分割し、バースの最後の名詞をライム対象の単語として抽出する。

次に、ライム対象の単語と同じ母音を持つ単語を、ライムの候補として検索する。検索には AWD-J core 辞書<sup>\*6</sup>に登録されている単語を使用する。AWD-J core 辞書は、日常で良く使用される単語を 15,220 件収録した辞書である。ラップバトルは話し言葉で構成される傾向があるため、日常的によく使用される語をライムの検索対象とした。辞書に登録されている単語はあらかじめ母音列に変換しており、対象となる単語と同じ母音列を持つ単語が検索される。検索制約を緩和するために、ライム対象の単語は末尾から 4 つ目の母音まで上限として考慮して、それらが一致する単語のみを検索した。ライム対象の単語の母音数が 4 未満の場合は、その数と同じ母音を末尾に持つ単語が検索される。

最後に、検索された単語の中から、ライム対象の単語と

<sup>\*6</sup> <https://sociocom.naist.jp/awd-j/>

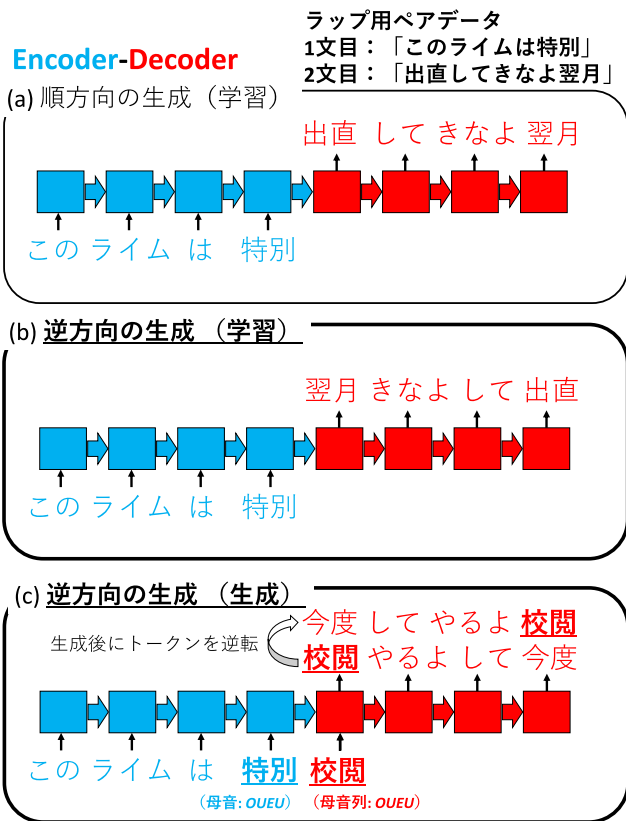


図 3 順方向の生成と逆方向の生成の違い

Fig. 3 Differences between forward generation and reverse generation.

意味的に関連する単語を選択する。バース内に含むライムは、ただ母音列が一致するだけでなく、単語間に意味的な類似があると評価が高い。そこで本研究では、ライムに意味類似性を考慮した。意味類似性の計算には、事前学習された fastText<sup>\*7</sup>の日本語の単語埋め込みを用い、ライム対象の単語とのコサイン類似度が高い順に上位7単語を選択する。さらに、生成されたラップ文のいずれかが相手のバースに対するアンサーとなるように、ライム対象の単語も追加し、合計8単語をライムとして使用する単語として選択する。

### 3.3 ラップ文生成

ライムとして使用する単語を8単語選択後、次はそれぞれの単語を含むラップ文を生成する。一般に、Decoderを用いて文を生成する場合、文の先頭から末尾にかけて単語が出力される(図3(a))。このようなモデルでは、任意の単語で終わるような文を生成することは困難である。そこで、本研究では、Yamadaら[10], [11]とXueら[16]の手法を参考に、文の末尾から先頭にかけて、逆向きに単語を出力して文を生成する。つまり我々の手法では、まず逆向きの文を用いてラップ文生成モデルを学習する(図3(b))。

<sup>\*7</sup> <https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.ja.300.vec.gz>

そして、学習したモデルを用いて、選択した単語を開始トークンとしてラップ文を生成する(図3(c))。

ラップ文生成には、BERT2BERT[2]を用いた。BERT2BERTはTransformerのEncoderとDecoderにBERT[3]の事前学習済みパラメータを読み込ませた生成モデルである。本研究では、東北大学から公開されている事前学習済みBERT<sup>\*8</sup>のパラメータを、BERT2BERTのEncoderとDecoderのパラメータの初期値とする。

BERT2BERTをファインチューニングするために、3.1節で作成したラップ文ペアデータを用いた。まず、図3(b)に示すように、事前学習モデルに対応したトークナイザを用いて、ペアの各文をトークンに分割する。次に、ペアの最初の文をEncoderの入力文として使用し、2番目の文はトークンを逆順に並べてから、Decoderの出力文として使用する。これにより、モデルは文を逆方向に生成することができる。EncoderとDecoderの学習には、以下のハイパーパラメータを使用した。

- Batch size : 32
- Optimizer : Adam [19]
- ロス関数 : Cross Entropy Loss
- 学習率 : 2e-7
- Dropout : 0.1
- 最大文長 (最大トークン数) : 128
- Early stopping : 10 patience

モデルを学習するために、ラップ文のペアデータを8:1:1の割合で訓練データ、検証データ、テストデータに分割した。学習は224エポックで終了した。

最後に、学習したモデルを用いて、相手のバースの最後の文をEncoderに入力し、3.2節で選択されたライム対象の単語1つをDecoderの初期トークンとして指定し、トークン列を生成する。生成されたトークン列を反転して連結することで、選択された8つのライム対象の単語ごとにラップ文が生成される。生成に使用したパラメータは以下のとおりである。

- 最大文長 (最大トークン数) : 30
- Top-k : 10
- Top-p : 0.95
- No-repeat-n-gram-size : 2

### 3.4 バースを考慮したラップ文の並べ替え

3.3節では、8つのラップ文が独立に生成されるため、それらを適切な順番に並べてバースを構成する必要がある。そのために、与えられた2つのラップ文が、バースの文脈上連続した文として適切かどうかを予測する隣接文予測タスクを、BERTモデルで学習する(図4)。正例としてラップ文ペアデータの文ペアを使用し、負例はラップ文ペ

<sup>\*8</sup> <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

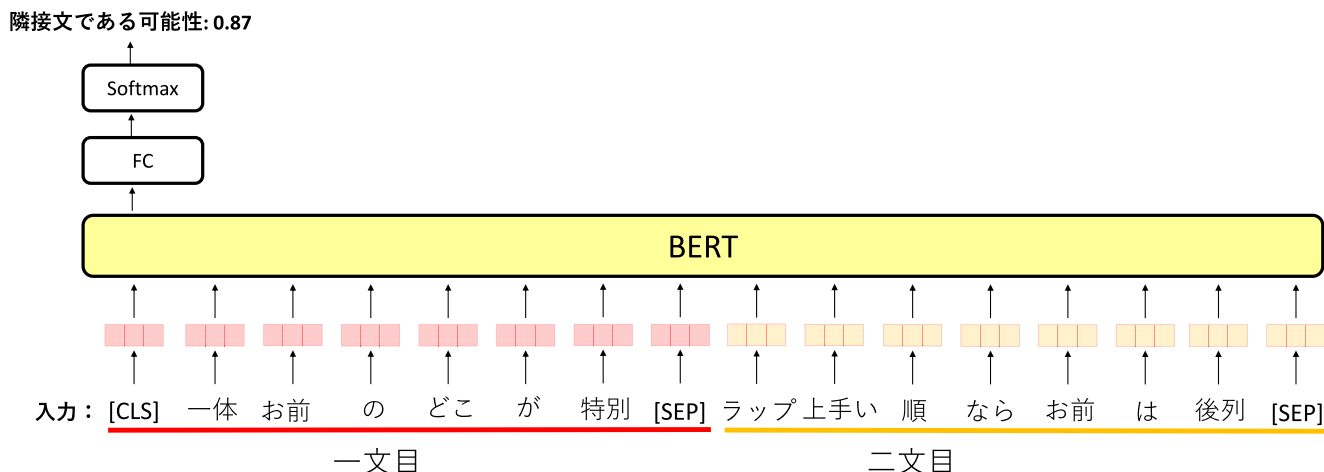


図 4 隣接文予測タスクの概要  
Fig. 4 Overview of the next sentence prediction task.

アデータの各文ペアの2番目の文を、ランダムに選んだ別のペアの第2文に置き換えることで作成した。学習データとして10,748組のペア(5,374組の正例と5,374組の負例)を作成し、8:1:1の割合で訓練データ、検証データ、テストデータに分割した。隣接文予測にも、東北大学が公開している日本語学習済みBERTモデル\*9を使用した。学習時のハイパーパラメータは以下のとおりである。

- Batch size : 8
- Optimizer : Adam [19]
- ロス関数 : Cross Entropy Loss
- 学習率 : 2e-7
- Dropout : 0.1
- 最大文長 (最大トークン数) : 128
- Early stopping : 5 patience

学習は30エポックで終了し、テストデータでの正解率は0.60であった。

学習したBERTモデルを使用して、ラップ文を並べ替えることでバースを構築する。本手法では、まず、相手のバースの最後に使われている単語を含むラップ文を、返答バースの最初の文として選択する。次に、学習済みBERTを使用して、残りの7文の中から、適切である確率が最も高い文を、バースの第2文として選択する。この選択プロセスを繰り返し、最も適切な残りの文をn番目の文の次のn+1番目の文として選択する(2 ≤ n ≤ 7)。結果として、8つの文からなる返答バースが構成される。

#### 4. 評価

提案手法の有用性を示すために定量評価と定性評価を行った。定量評価では、バースの自然さ、ライムの質、アンサーの質の3つの観点から評価を行った。定性評価では、ケーススタディとして、提案手法で生成した返答バースと比較手法で生成した返答バースを比較した。

スと比較手法で生成した返答バースを比較した。

#### 4.1 順方向の生成と逆方向の生成の比較

##### 4.1.1 定量評価

文末から逆方向に文を生成することの有用性を検証するために、文頭から順方向に文を生成する方法との比較を行った。比較手法のファインチューニング方法は、文の生成方向以外、3.3節と同様の方法を用いた。すなわち、3.3節と同様の方法を用いて文を順方向に生成するモデルを学習し(図3(a))、隣接文予測モデルにより文を並べ替えた。ただし、モデルの構造上、明示的なライムの生成は指定できない。ハイパーパラメータも同様とした。

返答バース生成のために、ラップバトルコーパスに含まれる1,287件のラップバトルの中からランダムに100件を選択し、各ラップバトルの最後の文を入力文とした。これら1,287件のすべてが学習に使用されたが、各ラップバトルの最後の文は、それに続く文が存在していないため、学習中に入力文として使用されていない未知のデータである。そして、各入力文に対して、提案方法と比較手法を用いて1つのバースを生成した。結果として、入力文100件に対して合計200件のバースが生成された。評価は、バースの自然さ、ライムの質、アンサーの質の3項目について、ラップバトル観戦歴5年以上の評価者2名による7段階評価(1:非常に悪い~7:非常に良い)を行った。

評価尺度はそれぞれが独立した評価となるように設計した。つまり、バースの自然さ、ライムの質、アンサーの質のそれぞれは、自身以外の2つの尺度から影響を極力受けないよう、以下のような教示により評価してもらった。

- バースの自然さは、バース内のラップ文の順番の自然さとラップ文の自然さ(文として破綻していないか)に基づいて評価する。
- ライムの質は、バース中にライムが含まれているか、

\*9 <https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking>

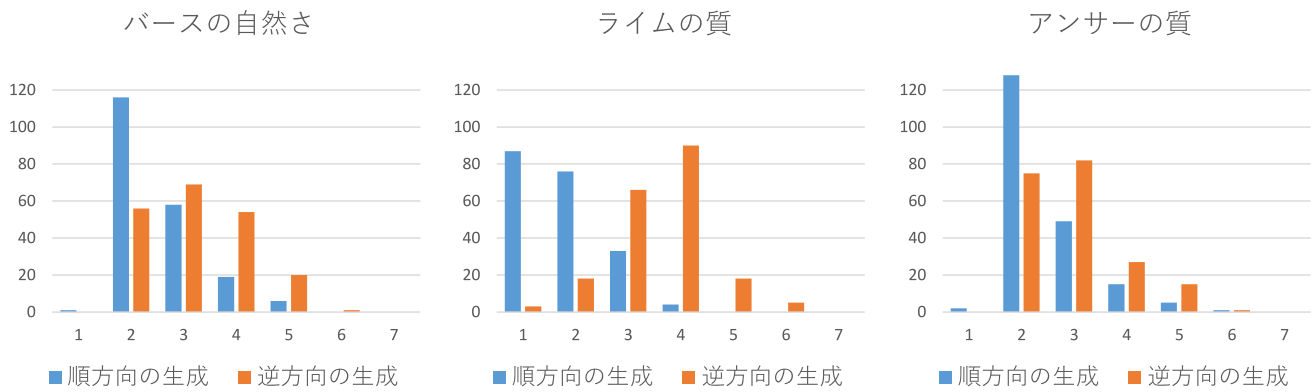


図 5 提案手法（逆方向の生成）と比較手法（順方向の生成）における 3 つの評価尺度のスコア分布。横軸は 7 段階の評価スコアであり、縦軸は 2 名の評価者による評価の総数である

Fig. 5 Score distribution for three evaluation metrics in the proposed method (reverse generation) and comparison method (forward generation). The horizontal axis represents a 7-point evaluation score, and the vertical axis represents the total number of evaluations by two evaluators.

表 3 提案手法（逆方向の生成）と比較手法（順方向の生成）における 3 つの評価尺度の平均スコア

Table 3 Average scores for three evaluation metrics in the proposed method (reverse generation) and comparison method (forward generation).

手法	バースの自然さ	ライムの質	アンサーの質
順方向の生成 (比較手法)	2.56	1.77	2.48
逆方向の生成 (提案手法)	<b>3.20</b>	<b>3.58</b>	<b>2.92</b>

ライムに意味類似性があるかを評価する。

- アンサーの質は、相手のバースに対して返答となっているかを評価する。

表 3 に、提案手法（逆方向の生成）と比較手法（順方向の生成）の 3 つの評価尺度の平均スコアを示す。表に示すように、提案手法はすべての評価尺度において比較手法を上回った。手法間の結果の違いをより詳細に分析するために、3 つの評価尺度のスコア分布を図 5 に示す。表 3 において、提案手法が比較手法をライムの質で大きく上回ったように、図 5 において、提案手法はライムの質のスコアが 4 以上のバースを多く生成し、比較手法はライムの質のスコアが 4 以上のバースをほとんど生成できなかった。これは順方向での生成において、ライムが生成されることが少なかったことなどが原因で、質の低下につながった可能性がある。また、提案手法と比較手法では、ライムの生成方向以外は同じであるため、表 3 のライムの質よりも、バースの自然さやアンサーの質の方が点数差が小さい。しかし、図 5 に示すように、比較手法では、「バースの自然さ」と「アンサーの質」において、スコア 2 のバースが多く生成されたのに対し、提案手法ではスコア 2 のバースが少なく、提案手法は低品質なバースが生成されにくいという利点があることが示された。これらの結果から、文末から

逆方向に文を生成することは有効であることが示された。

#### 4.1.2 定性評価

図 6 の上段と下段はそれぞれ提案手法と比較手法によって生成された返答バースの例である。上段の例では、相手のバースの中でライムに使用すべき単語の母音が赤色で強調されている。提案手法では、返答バースの各文が“iou”の母音を持つ単語で終わるようにすることで、ライムを考慮する。提案手法は、各文にライムを考慮するという制約を課しているにもかかわらず、意味のある文を生成することができ、返答バースとして適している。対照的に、比較手法の返答バースの文はどれも“iou”の母音を持つ単語で終わっていない。これらの結果からも、逆方向に学習して文を生成することの有用性が示された。

#### 4.2 バース単位での生成手法との比較

関連研究である DeepRapper [16] は、提案手法と同様にラップの歌詞データを逆方向に生成するモデルを学習することで、文末にライムを含んだラップを生成しているが、一方で、複数文で構成されたバース全体を生成単位としている点が提案手法と異なる。そのため、比較手法として DeepRapper にならない、本研究では「バース単位でラップを生成する手法」を用意した。具体的には、バース単位での生成モデルの学習方法は、3.3 節にて説明した提案手法のモデルの学習に準じているが、生成単位が文ではなくバースとなっている点が異なる。ただし、DeepRapper の実装に従い、文内の単語の生成順序は逆であるが、バース内の文の生成順序は元の順序とした。そのため、最初に生成する 1 文目の末尾にはライムを指定できるが、それ以降は文末にライムを指定することができない。また、学習時には、各バースの最後の文を入力文と見なして、その次のバース全体を正解データとして学習した。学習に用いたハイパーパラメータは以下のとおりである。



相手のバース： リズム倍で乗ってはめて遊ぶだけは俺もできる こんな奴のスキルならば更に上に行く ニガリを越すため3連覇 お前全然足りない鍛錬が おいおい俺にアンサーがない ってアンサーがないのは あんたじゃない勝ちたいなら韻 沢山踏もうコイツは f a k e 拡散希望(KIBOU)	提案手法で生成した返答バース： 俺のネタにしてもマジで腐って俺はここに拡散希望(KIBOU) そうじゃない俺はイケてるマジで政治志望(SIBOU) 何度も言ってる俺を否定してるんじゃないよみんなが見てきた理想(RISOU) 口だけは教えてくれると思ったんでなくて俺は否定して見てるだけの未来志向(SIKOU) 言ってる言っただけじゃねえ俺が持っているだけの魅力(MIRYOKU) そんなことは絶対に言わないって言われても興味はない俺の利用(RIYOU) ラップして確かに無限の意欲(IYOKU) お前が負けるのは俺は自分の意向(IKOU)
	比較手法で生成した返答バース： 俺が勝ち上がることはない 俺が勝つのを見ただけ 俺は俺が優勝した 俺も俺のやり方ではない 俺はそうやってるのは この場でやってるぜ 俺がラップはそのままやってきてるんだぜ 俺はラップは上手いんだ

図 6 提案手法（逆方向の生成）と比較手法（順方向の生成）によって生成された返答バースの例

Fig. 6 Examples of response verses generated by the proposed method (reverse generation) and comparison method (forward generation).

表 4 提案手法（文単位での生成）と比較手法（バース単位での生成）とテストデータ（ラップバトルコーパス）における 3 つの評価尺度の平均スコア

Table 4 Average scores for three evaluation metrics in the proposed method (sentence-level generation), comparison method (verse-level generation) and test data (rap battle corpus).

手法	バースの自然さ	ライムの質	アンサーの質
文単位での生成 (提案手法)	4.59	4.81	3.58
バース単位での生成 (比較手法)	2.50	1.70	1.65
参考	バースの自然さ	ライムの質	アンサーの質
テストデータ † (ラップバトルコーパス)	5.70	4.57	4.74

† ラッパーによる実際のバース

- Batch size : 8
- Optimizer : Adam [19]
- ロス関数 : Cross Entropy Loss
- 学習率 : 2e-7
- Dropout : 0.1
- 最大文長 (最大トークン数) : 512
- Early stopping : 10 patience

モデルの学習は 458 エポックで終了した。バースの生成に使用したパラメータは以下のとおりである。

- 最大文長 (最大トークン数) : 512
- Top-k : 10
- Top-p : 0.95
- No-repeat-n-gram-size : 2

本比較実験においては、実験条件を揃えるために提案手法を学習し直すこととし、ラップバトルコーパスに含まれる 1,287 件のバトルのうち、1,187 件を提案手法と比較手法

の学習データとして、残りの 100 件の中からランダムに 50 件を選んで、テストデータとして使用した。各テストデータを入力して生成した結果の評価は、バースの自然さ、ライムの質、アンサーの質の 3 項目について、4.1.1 項とは異なる評価者 2 名による 7 段階評価 (1 : 非常に悪い ~ 7 : 非常に良い) を行った。評価者は、各テストデータに対して提案手法あるいは比較手法が生成した結果のバースのテキストを見て評価した。さらに、参考のために、ラッパーによる実際のバースがどれくらい高く評価されるのかを調べる目的で、生成結果ではなく入力側のテストデータ 50 件に対しても、それぞれの 2 番目のバースを抜粋して、同様の 7 段階評価を行った。

#### 4.2.1 評価結果

表 4 に、提案手法による文単位での生成と、比較手法によるバース単位での生成、テストデータの 3 種類のバースに対する 3 つの評価尺度の平均スコアを示す。その結

<p>相手のバース：</p> <p>「どいてくれ」なんて確かに言っていない                  心の中で言ったお前の夢はさっきのVTRみたいに                  ここで終わる事になる                  これがリアル現実教えてやるぜ                  その身振り手振り止めた方が良い                  食わず嫌いの俺はT-TANGGとピータン</p>	<p>文単位での生成手法：</p> <p>今のお前のようにピータン                  こいつに勝てるだけじゃないんだなピーマン                  二十年前にこいつ殺しに来たサラリーマン                  お前がお前の飯を食ってここに来たって言ったのはおじいさん                  相手じゃねえ俺はおじいちゃん                  言いたいことがマジでじいちゃん                  お前の兄さん                  それを気にすんな俺は俺のガキのお兄ちゃん</p>
	<p>バース単位での生成手法：</p> <p>やるしかないぜここでピーマン                  後で見せるぜこの上で適当な首飛ばして                  それくらいの首って名前がつかばっか本戦でどっかでつか予選の予選予選で同じ連続連続で                  この前で韻踏むん上から下がらないからどんだ踏ん踏むだけだたっただ一回だけ韻踏んでくだけなら                  全国でその1度もお前こまで九州に一度一回も韻を踏んで踏んだ前の奴の上は負けない                  だから俺たちはお前のやり方で順番にやってきたそこの全国にここから先に                  同じような名古屋のローカルから全部ぶち上げてやるだけだ俺は愛知全部背負って行くぜ！                  これは大阪のバトルで決勝でお前さって愛知県から名古屋でいったらマジで                  全国全国の最年少でレギュラーで福岡まで来てるからこの場所でバトルは最初に                  その前前からその目を見てあんたなくたってそれ上上上がったこのバトルだけは                  何四回目目の前にお前はここでそれだけラップで一つ一つだけに九州の真ん中から                  しっかりと頭から全て吹き飛ばすために今日は次の挑戦してお前みたいなもんだ                  その次から3回戦から一回戦で1回戦負けたぜ                  長崎は1つだけのライブはいつものとこでやっとなんと長崎長崎のリベンジしようって話</p>

図 7 提案手法（文単位での生成）と比較手法（バース単位での生成）によって生成された返答バースの例

Fig. 7 Examples of response verses generated by the proposed method (sentence-level generation) and comparison method (verse-level generation).

果、提案手法による文単位での生成ではバースの自然さが 4.59、ライムの質が 4.81、アンサーの質が 3.58 となり、バース単位での生成でのバースの自然さが 2.50、ライムの質が 1.70、アンサーの質が 1.65 よりも、大きく上回っていた。ラップバトルコーパスのテストデータの評価では、バースの自然さが 5.70、ライムの質が 4.57、アンサーの質が 4.74 と想定どおり全体的に高かった。

比較手法によるバース単位での生成手法のスコアが低い原因を調査するために、提案手法と比較手法のそれぞれの生成結果を図 7 に示す。この図から分かるように、比較手法が生成したバースの文長が長いだけでなく、バースの内容も意味不明な語や破綻した文が含まれているため、「バースの自然さ」や「アンサーの質」が低くなったと考えられる。これは同じラップバトルコーパスを学習データとしているが、提案手法は文単位での生成であるため、52,422 文を学習できたのに対して、比較手法ではバース単位の生成であるため、6,791 バースと少ないデータで学習する必要があった。そのため、自然なバースを生成できなかつたと考えられる。

「ライムの質」に関しては、バース単位の生成手法では、入力バースの最後の文末「ピータン」に対して、生成されたバースの最初の文末では「ピーマン」とライムを考慮しているが、それ以外の文末ではライムを考慮できていない。これは、バース単位の生成手法では、逆方向での生成でバースの最初のライムは考慮できても、バース内の単語のライムを考慮できないのに対し、提案手法は文単位での生成でバース内のすべての文末にライムを含むことができているからである。そのため、表 4 で「ライムの質」だけ

は、人間が歌ったラップバトルコーパスよりも提案手法の方が上回っており、これは自明ではない結果で興味深い。また、このように提案手法で各文末にライムが含まれているときに、比較手法より「アンサーの質」でスコアが 1.93 高くなっている。前述の表 3 の結果でも、比較手法の「順方向の生成」より、各文末にライムが含まれる提案手法の「逆方向の生成」の方が「アンサーの質」でスコアが 0.46 高くなっており、その観点でも提案手法の有用性が分かる。

#### 4.2.2 Shimon [18] との比較

提案したラップ文の並べ替え手法の有用性を評価するために、関連する Shimon [18] の並べ替え手法と比較した。Shimon の並べ替え手法では、歌詞内のライムに着目した並べ替えを行っており、まず、ラップ文 1 文内で、最もライムを含んでいる文を最初の文として選択する。次に、残りの 7 文の中から、最初の文に対して、最もライムを含んでいる文を 2 文目の候補として選択する。この選択プロセスを繰り返し、最もライムを含んでいる残りの文を  $n$  番目の文の次の  $n + 1$  番目の文として選択する ( $2 \leq n \leq 7$ )。

この比較実験では、4.2 節で 7 段階評価をしたデータの中から、以下の 2 種類を再利用して、並べ替えの対象データとした。

- 提案手法で生成したバース
- ラップバトルコーパス内のバース

まず、提案手法で生成したラップ文を、提案手法と Shimon の手法のそれぞれで並べ替えて比較する。ただし、生成したラップ文の集合に文脈的なつながりがない可能性もあるため、文脈的なつながりがある、ラップバトルコーパスのバースも並べ替えて比較する。

表 5 バースの並べ替えによるバースの自然さの平均スコア

Table 5 Average scores for the naturalness of verses through verse reordering.

手法	ラップバトルコーパス	提案手法で生成したバース
提案手法	<b>4.18</b>	<b>3.14</b>
Shimon [18]	3.48	2.75

本比較では、バースの並び順が重要であるため、「バースの自然さ」のみを7段階評価した。その結果を表5に示す。この結果から、提案手法で生成したバースとラップバトルコーパスのいずれにおいても、提案手法の並べ替え手法の方が優れていることが分かった。これは、Shimon [18]の並べ替え手法ではライムのみに着目して文の意味を考慮できていないのに対し、提案手法ではラップバトルコーパスで学習したBERTで意味を考慮しながら並べ替えができていたためだと考えられる。

## 5. まとめ

本論文では、ラップバトルにおける、ライムとアンサーを考慮した返答バース生成手法を提案し、その有用性を検証した。今回は日本語のラップバトルコーパスを用いたが、提案手法自体は言語に依存しないため、今後は英語などの他の言語においても有用性を検証したい。また、返答バースをより柔軟に構築するために、手法を拡張する予定である。特に、日本語ラップ [20] では、ライムの母音の類似性だけでなく、子音の類似性 [21] も考慮することが望ましいため、今後の課題として取り組む予定である。最後に、1章で述べたように、本研究はラップバトルに馴染みのない人のラップ練習を支援することを目的としている。そのため、提案手法を搭載したラップバトル対話システムを開発し、学習支援における有用性を検証したい。

謝辞 本研究はJSPS科研費JP21H03775, JP21H03774, JP21H03554, JP22H03905の助成を受けたものです。ここに記して謝意を表します。

## 参考文献

[1] Venla, S.: Interactive oral composition: Resources, strategies, and the construction of improvised utterances in a Finnish freestyle rap battle, *The Journal of American Folklore*, Vol.132, No.523, pp.3–35 (2019).

[2] Sascha, R., Shashi, N. and Aliaksei, S.: Leveraging pre-trained checkpoints for sequence generation tasks, *The Journal of Transactions of the Association for Computational Linguistics*, Vol.8, No.1, pp.264–280 (2020).

[3] Devlin, J., Chang, M., Lee, K. and Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding, *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL 2019*, pp.4171–4186 (2019).

[4] Mibayashi, R., Yamamoto, T., Tsukuda, K., Watanabe, K., Nakano, T., Goto, M. and Ohshima, H.: Verse gen-

eration by reverse generation considering rhyme and answer in Japanese rap battles, *Proc. 16th International Symposium on Computer Music Multidisciplinary Research, CMMR 2023*, pp.30–41 (2023).

[5] Rumelhart, D., Hinton, G. and Williams, R.: Learning internal representations by error propagation, Technical Report, University of California, San Diego (1985).

[6] Gers, F., Schmidhuber, J. and Cummins, F.: Learning to forget: Continual prediction with LSTM, *The Journal of Neural Computation*, Vol.12, No.10, pp.2451–2471 (2000).

[7] Sutskever, I., Vinyals, O. and Le, Q.: Sequence to sequence learning with neural networks, *Proc. 27th International Conference on Neural Information Processing Systems, NIPS 2014*, pp.3104–3112 (2014).

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, U. and Polosukhin, I.: Attention is all you need. *Proc. 31st International Conference on Neural Information Processing Systems, NIPS 2017*, pp.6000–6010 (2017).

[9] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I.: Language models are unsupervised multitask learners, Technical Report, OpenAI (2019).

[10] 山田康輔, 人見雄太, 田森秀明, 岡崎直観, 乾健太郎: 指定語句を確実に含む見出し生成, 言語処理学会第27回年次大会, 言語処理学会2021, pp.1070–1074 (2021).

[11] Yamada, K., Hitomi, Y., Tamori, H., Sasano, R., Okazaki, N., Inui, K. and Takeda, K.: Transformer-based lexically constrained headline generation, *Proc. 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pp.4085–4090 (2021).

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Proc. 26th International Conference on Neural Information Processing Systems, NIPS 2013*, pp.1–9 (2013).

[13] Potash, P., Romanov, A. and Rumshisky, A.: Ghost-Writer: Using an LSTM for automatic rap lyric generation, *Proc. 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pp.1919–1924 (2015).

[14] Malmi, E., Takala, P., Toivonen, H., Raiko, T. and Gionis, A.: DopeLearning: A computational approach to rap lyrics generation, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2016*, pp.195–204 (2016).

[15] Manjavacas, E., Kestemont, M. and Karsdorp, F.: Generation of hip-hop lyrics with hierarchical modeling and conditional templates, *Proc. 12th International Conference on Natural Language Generation, INLG 2019*, pp.301–310 (2019).

[16] Xue, L., Song, K., Wu, D., Tan, X., Zhang, N., Qin, T., Zhang, W. and Liu, T.: DeepRapper: Neural rap generation with rhyme and rhythm modeling, *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021*, pp.69–81 (2021).

[17] Wu, D. and Addanki, K.: Learning to rap battle with bilingual recursive neural networks, *Proc. 24th International Joint Conference on Artificial Intelligence, IJCAI 2015*, pp.2524–2530 (2015).

[18] Savery, R., Zahray, L. and Weinberg, G.: Shimon the rapper: A real-time system for human-robot interactive rap battles, *Proc. 11th International Conference*

on Computational Creativity, ICCV 2020, pp.212-219 (2020).

- [19] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *Proc. 3rd International Conference on Learning Representation, ICLR 2015* (2015).
- [20] Manabe, N.: Globalization and Japanese creativity: Adaptations of Japanese language to rap, *The Journal of Ethnomusicology*, Vol.50, No.1, pp.1-36 (2006).
- [21] Kawahara, S.: Half rhymes in Japanese rap lyrics and knowledge of similarity, *The Journal of East Asian Linguistics*, Vol.16, No.2, pp.113-144 (2007).



三林 亮太 (学生会員)

2022年兵庫県立大学大学院応用情報科学研究科博士前期課程修了。修士(応用情報科学)。現在、兵庫県立大学大学院情報科学研究科博士後期課程在学中。日本データベース学会学生会員。



山本 岳洋 (正会員)

2011年京都大学大学院情報学研究科博士後期課程修了。博士(情報学)。現在、兵庫県立大学大学院情報科学研究科/社会情報科学部准教授。主に情報検索におけるユーザインタラクションやユーザ理解に関する研究に従事。

日本データベース学会, ACM, 電子情報通信学会各会員。



佃 洸撰 (正会員)

2014年京都大学大学院情報学研究科博士後期課程修了。博士(情報学)。現在、産業技術総合研究所主任研究員。主に音楽コンテンツおよびユーザ生成コンテンツを対象とした情報推薦・情報探索の研究に従事。日本データベース学会会員。

ス学会会員。



渡邊 研斗 (正会員)

2018年東北大学大学院情報科学研究科博士後期課程修了。現在、産業技術総合研究所人間情報インタラクション研究部門主任研究員。自然言語処理, 音楽情報処理, 歌詞情報処理の研究に従事。言語処理会員。



中野 倫靖 (正会員)

2008年筑波大学大学院図書館情報メディア研究科博士後期課程修了。博士(情報学)。現在、産業技術総合研究所主任研究員。日本音響学会会員。2009年情報処理学会山下記念研究賞(音楽情報科学研究会), 2013年 Sound

and Music Computing Conference (SMC 2013) The Best Paper Award, 2018年 IEEE Pacific Visualization Symposium (PacificVis 2018) Honorable Mention Poster Award 等各受賞。



後藤 真孝 (正会員)

1998年早稲田大学大学院理工学研究科博士後期課程修了。博士(工学)。現在、産業技術総合研究所首席研究員。2009~2017年にIPA 未踏 IT 人材発掘・育成事業 PM, 2016~2022年に JST ACT-I「情報と未来」研究総

括を兼任。現在、JST 創発 PO, 日本学術会議連携会員, 統計数理研究所客員教授, 筑波大学大学院教授(連携大学院)等を兼任。日本学士院学術奨励賞, 日本学術振興会賞, ドコモ・モバイル・サイエンス賞基礎科学部門優秀賞, 市村学術賞, FIT 船井業績賞, 科学技術分野の文部科学大臣表彰若手科学者賞, 星雲賞等, 65件受賞。本会フェロー。



大島 裕明 (正会員)

2007年京都大学大学院情報学研究科博士後期課程修了。博士(情報学)。現在、兵庫県立大学大学院情報科学研究科/社会情報科学部准教授。主に情報検索, ソーシャルコンピューティング, デザイン学の研究に従事。電子情

報通信学会, ACM, 日本データベース学会各会員。

(担当編集委員 白井 清昭)