

# ラップバトルにおけるライムとアンサーを考慮した 逆向き生成によるバース生成

三林 亮太<sup>†</sup> 山本 岳洋<sup>††</sup> 大島 裕明<sup>†,††</sup>

<sup>†</sup> 兵庫県立大学 大学院応用情報科学研究科 〒650-0047 兵庫県神戸市中央区港島南町 7-1-28

<sup>††</sup> 兵庫県立大学 大学院情報科学研究科 〒651-2197 兵庫県神戸市西区学園西町 8-2-1

E-mail: †{aa20r511,ohshima}@ai.u-hyogo.ac.jp, ††t.yamamoto@sis.u-hyogo.ac.jp

**あらまし** 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バース生成手法を提案する。ラップバトルとは、2名のラッパーが即興ラップを交互におこない、どちらのラップが優れているかを競う競技である。ラップバトルでおこなわれる即興ラップのことをバースと呼び、バースにはライム（韻）とアンサー（返答）を含む必要がある。ライムとは、ある単語に対して母音列が同一の語のことであり、アンサーとは、相手の発言をふまえた返答をすることである。このような、ライムとアンサーを考慮したバース生成手法はこれまでに提案されていない。そこで、本研究では、ラップバトルにおけるライムとアンサーを考慮したバース生成手法を提案する。提案手法では、生成と選択の2つのステップにより、バースを構成する。まず、生成ステップでは、ライムを考慮したラップの生成をおこなう。GPT-2やBERT2BERTなどの、文書生成モデルを用いて、文末から文を生成する逆向きの生成により、文末にライムを考慮したラップを生成する。次に、選択ステップでは、アンサーを考慮したラップの選択をおこなった。生成したラップの中から、相手の発言をふまえたラップを、BERTの隣接文予測を用いて選択することで、アンサーを考慮したラップをバースに含む。以上の、生成と選択の2つのステップにより、ライムとアンサーを考慮したバース生成をおこなった。

**キーワード** ラップバトル, 自然言語処理, BERT, GPT-2

## 1 はじめに

ラップバトルとは2名<sup>1</sup>のラッパーが即興ラップ（バース）を交互におこない、どちらのラップが優れているかを競う競技である。たとえば、図1に示すように、先攻のラッパーの「お前はできないライム作成/俺は忙しい博士学生」というバースに対して、後攻のラッパーは「博士学生ならば院で覚醒しろ/はっきり決めようぜ黒か白」という返答のバースを返す。これを数回繰り返して、どちらのラップが優れていたかを競う。

バースには、ライム（韻）とアンサー（返答）を含む必要がある<sup>2</sup>。ライムとは直感的には、ある単語に対して母音列が同一の語のことであり、アンサーとは、相手の発言をふまえた返答をすることである。たとえば、図1のバース2に示す「博士学生ならば院で覚醒しろ」という発言は、相手の発言であるバース1の「俺は忙しい博士学生」をふまえて返答しているため、アンサーといえる。

このような、ライムとアンサーを考慮したバース生成手法は、著者らの知る限り存在しない。ラップバトルにおけるバース生成の研究としては、Wuら[15]とSaveryら[10]と三林ら[18]の3件しかなく、これらの研究では、ライムとアンサーは同時

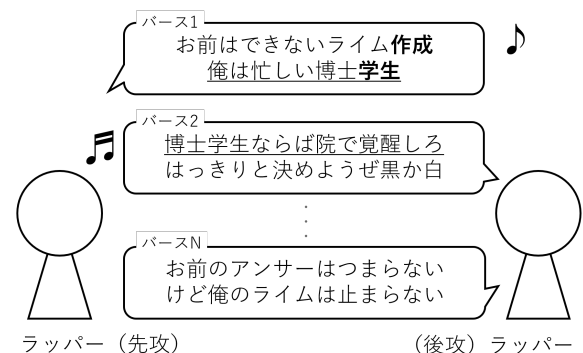


図1 ラップバトルの概要図

に考慮されていない。

そこで、本研究では、ラップバトルにおけるライムとアンサーを考慮したバース生成手法を提案する。本研究では、言語生成モデルであるBERT2BERT [1], [9], RoBERTa2RoBERTa [3], [9], GPT-2 [8]を用いたバース生成手法を提案する。提案手法では、生成と選択の2つのステップにより、バースを構成する。まず、生成ステップでは、バースを入力に、ライムを考慮したラップの文を生成する。ラップの文を生成する際には、文を逆向きに生成することで、ライムを末尾に考慮したラップの文生成をおこなう。選択ステップでは、生成したラップの文をランキング手法により選択する。選択方法には、入力バースに対して、アンサーとなるようなラップ文の選択と、バース内の文脈のつながりを考慮したラップ文の選択をおこなう。選択した

1: 3対3など、複数人でおこなう場合もあるが、1対1の2名が主流である。

2: バトル開始時の先攻の最初のバースを除く。

ラップ文を組み合わせることで、ライムとアンサーを考慮したパースを得る。

本論文では、まず、ラップバトルにおけるパース生成に関連する研究について2節で述べる。次に、3節と4節で、ラップバトルにおけるパース生成の問題定義をおこない、パース生成手法と実験設定について詳細に述べる。最後に、5節と6節で、パース生成手法による結果と考察を述べ、まとめと今後の課題について述べる。

## 2 関連研究

本研究はおおきく、ライムとアンサーという制約付きの文書生成問題といえる。関連する研究としては、一般的な文書生成手法に加えて、ライムを考慮した文書生成やアンサーを考慮した対話生成などがあげられる。

### 2.1 文書生成に関する研究

近年の文書生成手法は、深層学習を用いた生成が一般的である。これまでに、深層学習を用いた文書生成には、RNN や LSTM といった自己回帰型の構造を用いた seq2seq モデル [11] が用いられてきた。seq2seq モデルは文書情報を集約する Encoder と集約した情報から文書を生成する Decoder を持つモデルであり、Encoder-Decoder と呼ばれる。その後、Attention と呼ばれる機構が提案され、文書情報をうまく選択しながら生成することが可能となった。近年、Attention を用いた、Transformer と呼ばれるモデルが提案されている。Transformer は、大規模なパラメータと Attention を用いた Encoder-Decoder モデルである。Transformer を用いた有名なモデルとして、BERT と GPT-2 があげられる。BERT [1] とは、汎用的に用いられる自然言語文書の特徴抽出器である。BERT は、Transformer [13] の Encoder のみを複数層重ねたモデルであり、特徴としては、文書の情報を双方向から考慮できるため、前後の文脈が考慮できる特徴がある。また、BERT の改良手法として RoBERTa [3] が存在する。GPT-2 とは、生成タスクで汎用的に用いられる、文書生成モデルである。GPT-2 [8] は、Transformer の Decoder のみを複数層重ねたモデルである。特徴としては、BERT の双方向からの情報を考慮する機構とは異なり、単方向の情報しか見ない。これらのモデルは、事前に大量のコーパスで学習をおこない、汎用的な言語知識を獲得した状態で使用することが一般的である。事前学習したモデルに対して、特定のタスクに対する学習である、ファインチューニングをおこなうことで、タスクに特化したモデルを学習する。この事前学習したパラメータを用いて、BERT を Transformer の Decoder として用いる BERT2BERT と呼ばれる手法も提案されている。BERT2BERT [9] とは、BERT モデルの事前学習したパラメータを Transformer の Encoder と Decoder に転移した生成モデルである。BERT モデルで得た事前学習の重みを用いることが、Rothe ら [9] の生成タスクにおいて、有効であることが示されている。同研究では、RoBERTa を用いた、RoBERTa2RoBERTa も提案されている。

### 2.2 ラップバトルにおけるパース生成に関する研究

ラップバトルにおけるパース生成に関する研究は少ない。Wu らによるラップバトルのチャットボットシステムでは、RAAM (Recursive Auto-Associative Memory) を改良した TRAAM (Transduction Recursive Auto-Associative Memory) と確率的モデルを用いて、パースの生成に取り組んだ [15]。結果として、TRAAM の生成結果が良いことを示した。Shimon らはインタラクティブなラップバトルシステムを作成し、返答速度に注目し生成をおこなっている [10]。生成手法には LSTM を使用している。三林らはテンプレートと Transformer を用いた生成をおこなった [18]。これらは既存のパースに対して、相手の発言を考慮した語を置き換えることで、パースの生成をおこなっている。しかし、単語の置き換えにより、文書が破綻しやすくなる傾向にあった。筆者らの知る限り、ラップバトルにおけるパース生成に関する研究は以上の3つである。これらの研究では、ライムとアンサーは同時に考慮されていない。

### 2.3 ライムを考慮した文書生成に関する研究

ライムを考慮した文書生成手法は、ラップの生成や歌詞の生成において、いくつか提案されている。Potash らは、LSTM モデルを用いて、既存のラップ歌詞データから、新たなラップを生成する、Ghostwriter と呼ばれる手法を提案している [7]。Malmi らは、生成したラップの文をひとつ前の文との類似度を比較することで、次の文をもっともらしいものを選択する、DopeLearning と呼ばれる手法を提案している [4]。Xue らは、Transformer-Decoder を用いて、ビートを考慮したラップ生成をおこなう、DeepRapper と呼ばれる手法を提案している [16]。Nikolov らは、Rapformer と呼ばれる手法を提案しており、seq2seq [12] モデルを用いて、既存のテキストからラップ調にスタイル変換をおこなっている [6]。Manjavacas らは、LSTM を用いて、歌詞としてのラップの生成をおこなっている [5]。以上のように、歌詞としてのラップ生成はいくつか提案されており、近年は深層学習モデルによる生成がほとんどである。

### 2.4 アンサーを考慮した対話生成に関する研究

アンサーを考慮した対話生成には、いくつか関連する研究がある。Wolf らが提案した TransferTransfo [14] では、ペルソナ情報を考慮した対話生成手法を提案している。ラップバトルにおいて、ユーザのペルソナ情報を元にアンサーを生成する方法は一般的におこなわれているため、本研究と関連が深い。また、稲葉らは2ちゃんねるの対話データを収集し、NoneJupiterBot [17] と呼ばれる対話システムを提案している。稲葉らが作成した、おーぶん2ちゃんねる対話コーパスには、いくつかラップバトルに似た、他人の痛いところを突くような表現が含まれており、本研究と関連する部分がある。

以上のように、ラップバトルにおけるパース生成に関する研究は、まだ十分におこなわれておらず、ライムをアンサーを同時に考慮する生成手法は提案されていない。個別のタスクだと、ライムを考慮した生成手法では、ラップ生成手法や歌詞生成手法などがいくつか提案されており、現在は、Transformer

をベースとした深層学習モデルによる生成が主流である。アンサーを考慮した生成では、ペルソナ情報を考慮した対話生成や、2ちゃんねるのテキスト情報を元にした対話生成が存在する。

### 3 ラップバトルにおけるバース生成

#### 3.1 ラップバトルコーパス

本研究をおこなうにあたって、ラップバトルを文字起こししたテキストデータである、ラップバトルコーパスを作成した。ラップバトルに関するテキストデータは公開されていないため、独自にデータを収集し、ラップバトルコーパスの作成をおこなった。ラップバトルコーパスの作成はクラウドソーシングサイト Lancers<sup>3</sup>を利用してクラウドソーシングをおこなった。クラウドソーシングの内容は、YouTube上に存在するラップバトル動画内のバースを文字起こしをしてもらうものである。対象となるラップバトル動画は以下の3つのYouTubeチャンネルである。

- UMB<sup>4</sup>
- 凱旋 MCBattle<sup>5</sup>
- 戦極 MCBATTLE<sup>6</sup>

ワーカーにはこちらが作成した作業用サイトにアクセスしてもらい、指定するラップバトル動画のリンク先のYouTube上のラップバトル動画に対して文字起こしをもらった。

また、ラップバトルの番組内のラップバトルについても、文字起こしをおこなった。対象としたラップバトルの番組は以下の2つである。

- 高校生ラップ選手権
- フリースタイルダンジョン

これらのラップバトル番組についてはクラウドソーシングを利用せず、独自に文字起こしをおこなった。本研究では、合計1,287件のラップバトルの文字起こしコーパスを作成した。作成したラップバトルコーパスの統計量を表1に示す。ラップバトルコーパスに対する前処理として、アルファベットと数字はすべて全角に変換し、記号はすべて削除した。

##### 3.1.1 アンサーペアデータ

ラップバトルコーパスを元に、アンサーペアデータを作成した。アンサーペアデータとは、ラップバトルにおける、あるバースに対して、返答バースのアンサーをペアにしたデータである。たとえば、図1に示すラップバトルを例に挙げると、バース1の「お前はできないライム作成/俺は忙しい博士学生」と、バース2に示す「博士学生ならば院で覚醒しろ」をペアとするデータである。今回、アンサーは返答バースの一番最初の文に設定し、アンサーペアデータを作成した<sup>7</sup>。

#### 3.2 問題定義

本研究における問題定義は以下の通りである。

表1 ラップバトルコーパスの統計量

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バースあたりのバースセンテンスの平均	7.71
1バースセンテンスあたりのトークン数の平均	15.40

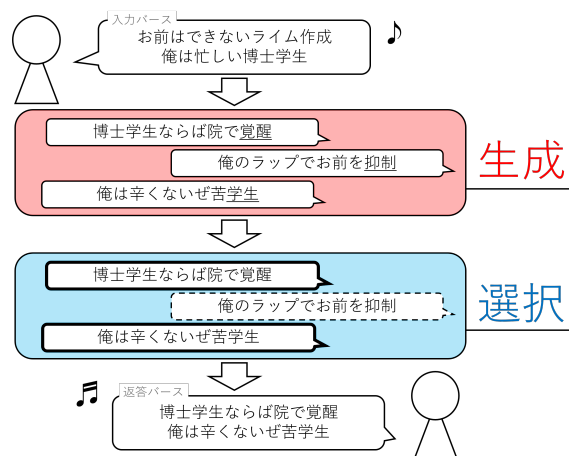


図2 提案手法の概要図：提案手法は、生成と選択の2つのステップによって構成される。ステップ1では、ライムを考慮したラップ生成をおこなない、ステップ2では、アンサーを考慮したラップ選択をおこなう。

- 入力：バース
- 出力：ライムとアンサーを含む返答のバース

本研究で定義するバースは、テキストで書かれたラップ調の文書である。たとえば、図1で示した、「お前はできないライム作成/俺は忙しい博士学生」などがバースである。本研究では、このようなバースを1つ入力すると、返答のバース（返答バース）を1つ出力する。

出力する返答バースには、ライムとアンサーを含む必要がある。本研究においてライムは、モーラが一致する語のペアと定義する。モーラとは、俳句の5・7・5などの音を数えるときによく用いられる、音の長さの単位である。具体例をあげると、「マイク」という言葉は、「マ・イ・ク」の3モーラと数えられる。モーラは、促音の「っ」や長音の「ー」を1モーラとして数える。たとえば、「ラッパー」は「ラ・ッ・パ・ー」の4モーラと数えられる。本研究では、このモーラが一致する語のペアがバース内に含まれる際に、ライムを考慮していると定義する。また、本研究では、文末のライムを生成の対象とする。ライムは文中の任意の箇所に含んでよいが、ラップにおいては、文末の末尾にライムを含むことが基本的であり、ラップバトルにおいても、同様のことが言える。よって、本研究では、文末の末尾にライムを含んだラップの生成をおこなう。

アンサーとは、相手の発言をふまえた返答をすることである。たとえば、図1のバース2に示す「博士学生ならば院で覚醒しろ」という発言は、相手の発言であるバース1の「俺は忙しい博士学生」をふまえて返答しているため、アンサーである。このような返答がバース内に存在する場合、アンサーを含むとい

3: <https://www.lancers.jp/>

4: <https://www.youtube.com/user/umbofficial>

5: [https://www.youtube.com/channel/UCe\\_EvY8GrvYgx8PbWbC75g](https://www.youtube.com/channel/UCe_EvY8GrvYgx8PbWbC75g)

6: <https://www.youtube.com/user/senritumc>

7: ラップバトルでは、アンサーをバースの最初に含むことが多いため。

## Encoder-Decoder

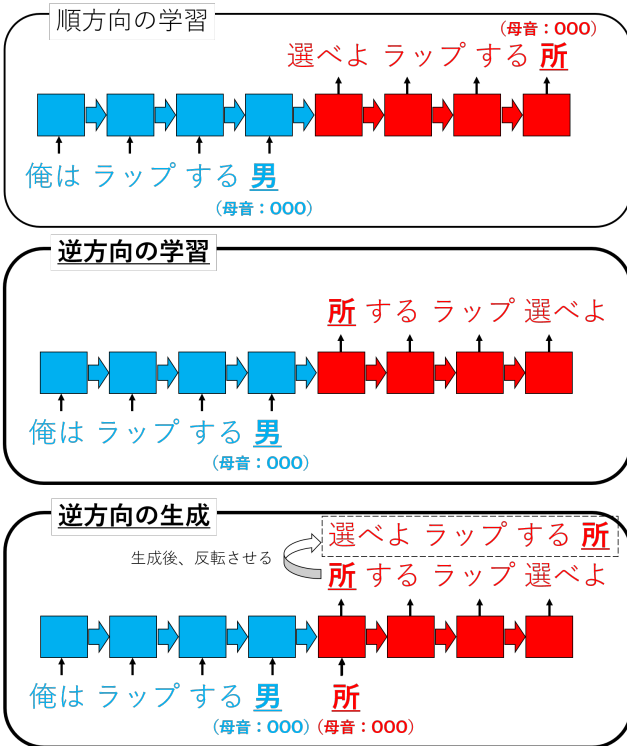


図 3 逆方向生成の概要

う。アンサーの定義はあいまいな部分が多いため、人手による判断で、アンサーの判別をおこなう。

### 3.3 パース生成手法の概要

パース生成手法は、図 2 に示すように、生成と選択の 2 つのステップで構成される。生成のステップでは、パースを入力に、文書生成モデルを用いて、ライムを考慮したラップを複数生成する。選択のステップでは、生成した複数のラップの中から、アンサーとして適しているラップを BERT モデルを用いて選択する。その後、選択した文をパースの文脈を考慮した並べ替えをおこなうことで、ライムとアンサーを考慮したパースを生成する。

### 3.4 生成

本節では、逆方向生成による、ライムを考慮したラップ生成手法について述べる。提案手法では、末尾にライムを考慮したラップを生成する。ラップの生成には、通常の順方向の生成とは異なる、逆方向からの文生成をおこない、ライムを考慮したラップを生成する。

ラップの生成には、Transformer Encoder-Decoder<sup>8</sup>を用いる。Encoder-Decoder は、Encoder 部分で情報を集約し、Decoder 部分で文書生成をおこなう。Decoder はモデルの構造上、左から右への順方向にしか生成できない。しかし、この順方向の生成では、末尾にライムを考慮することが難しい。そこで、本研究では、文を末尾から生成する、逆方向生成を提案する。文を逆方向に生成するようにモデルを学習し、最初に生成する

語をライムに指定することで、末尾にライムを考慮した生成がおこなえる。

逆方向生成とは、文を末尾のトークンから生成する方法である。通常の文生成では、図 3 の最上部に示すように、左から右の順番で文を生成する。逆方向生成では、図 3 の最下部に示すように、語の生成順番を逆にして生成する。最終的な出力を得るには、生成したトークン列を反転させることで、順方向の文を得る。この方法により、Decoder の構造を変えることなく、末尾から、ライムを考慮した生成が可能となる。

逆方向生成の具体的な流れを説明する。まず、図 3 の中部に示すように、入力文と出力文をトークナイザでトークンレベルに分割する。入力文はそのまま、出力文のみをトークン順を逆転させたペアデータを作成する。たとえば、「俺は/ラップ/する/男」という入力文に対して、「所/する/ラップ/選べよ」という出力文がペアになったペアデータが得られる。この入力文から、出力文を生成するようにモデルを学習することで、逆順の出力文を生成するようなモデルを構築する。

実験では、順方向でのラップ生成と、逆方向のラップ生成を比較することで、逆方向生成において、トークンを逆方向に学習できているかを検証する。今回使用するモデルは、Decoder を共通して持つが、モデルによって、入力形式や特殊トークンの扱いが異なる。それぞれのモデルに対する特殊トークンの処理や、学習方法の違いについては、4 節で詳しく述べる。

### 3.5 選択

本節では、ラップの文を選択する手法を 2 つ提案する。1 つ目は、アンサーを含むラップの文を選択する手法である。これは、パースを入力に、パースに対してアンサーとなるラップの文を選択する。2 つ目は、パースの文脈を考慮したラップの文を選択する手法である。これは、ラップの文を入力に、次のラップの文として文脈が自然なラップの文を選択する。以上の 2 つの手法を BERT を用いた隣接文予測にてランキング付けする。

#### 3.5.1 BERT を用いた隣接文予測

BERT には事前学習として、隣接文予測という学習方法が採用されている。隣接文予測とは、図 4 に示すように文のペアを入力に、それらが隣接する文として適切であるかを予測するタスクである。たとえば、「今日は一日中雨だった。」という 1 文目と「なので洗濯物が干せなかった。」という 2 文目のペアは隣接文であると予測されるのに対して、「今日は一日中雨だった。」という 1 文目と「私はラップが好きである。」という 2 文目のペアは隣接文とは予測されない。

隣接文予測は、BERT から出力されるベクトルに対して、クラス分類をおこなう。その際に使用する、ベクトルの値を用いてランキングがおこなえる。その値を元に文を並べ替えることで、複数の文ペアに対してのランキングが得られるため、隣接文予測はランキング手法ととらえることができる。

#### 3.5.2 アンサーランキング

アンサーランキングは隣接文予測によってラップ文のランキングをアンサーを含んでいる確率が高い順に出力する手法である。アンサーランキングの入力はパースであり、出力はラップ

8: 後述する、GPT-2 は Decoder のみ用いたモデルである。

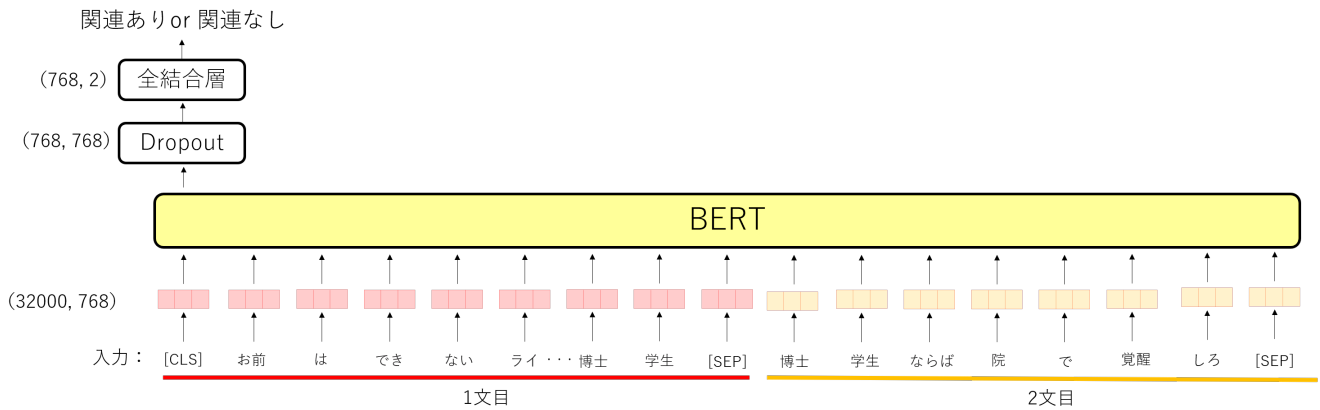


図 4 隣接文予測のモデル図

文のランキングである。図 4 に示すように、バースを入力し、ラップ文がアンサーを含むかを判定する隣接文予測を学習する。

### 3.5.3 ラップ文ランキング

ラップ文ランキングは隣接文予測によって、ラップ文のランキングを入力した文に対して、隣接文として自然であるかをランキングする手法である。ラップ文ランキングの入力は、ラップ文であり、出力は、続きの文として自然であるラップ文である。図 4 に示すように、あるラップ文を入力し、ラップ文が隣接する文であるかを判定する隣接文予測を学習する。

## 4 実 験

### 4.1 ライムを考慮したラップ生成

本節では、ライムを考慮したラップ生成の実験設定について述べる。ラップ生成に用いる、文書生成モデルは以下の 3 つを用いた。

- GPT-2
- BERT2BERT
- RoBERTa2RoBERTa

学習するコーパスは、アンサーペアを用いる。アンサーペアデータは入力バースとアンサーの文がペアになったデータである。本実験では、アンサーペアデータを用いて、通常の生成と逆向き生成をおこなった。逆向き生成では、アンサーペアデータ内の、アンサー文をトークン化した後に、反転させる前処理をおこなった。各モデルの学習に用いた、ハイパーパラメータはどちらの生成方向においても同じパラメータを用いた。

#### 4.1.1 GPT-2

GPT-2 モデルは、Transformer の Decoder を用いた生成モデルである。今回用いた GPT-2 モデルは rinna 社の事前学習モデル<sup>9</sup>を使用した。rinna 社の事前学習済み GPT-2 モデルは日本語 Wikipedia データと、日本語 CC-100 データセットを用いて事前学習されている。

GPT-2 のファインチューニングには、アンサーペアデータを用いた。バースとアンサー文を特殊トークンである「<s>」と「</s>」を用いて区切った。具体的には、「<s>バース</s>ア

ンサー文</s>」のような形式で入力した。また、文長が短い場合は右詰めで、特殊トークンである「<pad>」で埋める処理をおこなった。学習時のロス計算には、<pad>トークンは含まない。ハイパーパラメータとロス関数は表 2 に示す。学習結果は、287 エポックで終了した。

#### 4.1.2 BERT2BERT

BERT2BERT モデルは、BERT の重みを Transformer の Encoder-Decoder に転移させたモデルである。今回、Encoder と Decoder とともに、東北大学の事前学習済み BERT<sup>10</sup>を用いた。東北大学の事前学習済み BERT モデルは、日本語 Wikipedia のテキストデータで事前学習されている。

BERT2BERT のファインチューニングには、アンサーペアデータを用いた。入力形式は、バースの先頭に「[CLS]」トークンを付与し、末尾に「[SEP]」トークンを付与した。アンサー文も同様にトークンを付与した。文長が短い場合は、「[PAD]」トークンで埋める処理をおこなった。ロス計算の際には、「[PAD]」トークンは含まない。Encoder にバースを入力し、Decoder でアンサー文を生成する学習をおこなった。Encoder-Decoder とともに学習をおこない、パラメータを更新した。ハイパーパラメータとロス関数は表 2 に示す。学習結果は、196 エポックで終了した。

#### 4.1.3 RoBERTa2RoBERTa

RoBERTa2RoBERTa モデルは、BERT2BERT と同様に、RoBERTa の重みを Transformer の Encoder-Decoder に転移させたモデルである。今回用いた RoBERTa の事前学習モデルは、rinna 社の事前学習モデル<sup>11</sup>を使用した。事前学習モデルは Encoder-Decoder とともに、rinna 社の事前学習モデルを用いた。rinna 社の事前学習済み RoBERTa モデルは日本語 Wikipedia データと、日本語 CC-100 データセットを用いて事前学習されている。

RoBERTa2RoBERTa のファインチューニングは、BERT2BERT と同条件でおこなった。ハイパーパラメータとロス関数は表 2 に示す。学習結果は、177 エポックで終了した。

9 : <https://huggingface.co/rinna/japanese-gpt2-medium>

10 : <https://github.com/cl-tohoku/bert-japanese>

11 : <https://huggingface.co/rinna/japanese-roberta-base>

表 2 ハイパーパラメータ

項目	GPT-2	BERT2BERT	RoBERTa2RoBERTa	BERT (隣接文予測)
batch size	16	16	16	8
optimizer	Adam [2]	Adam [2]	Adam [2]	Adam [2]
ロス関数	CrossEntropy	CrossEntropy	CrossEntropy	CrossEntropy
学習率	2e-7	2e-7	2e-7	2e-7
Dropout 率	0.1	0.1	0.1	0.1
最大入力長	128	128	128	512
early stopping	3 patience	3 patience	3 patience	10 patience

## 4.2 アンサーを考慮したラップ選択

アンサーを考慮したラップ選択では、BERT を用いた隣接文予測をおこなった。BERT は東北大学乾研究室が提供する、日本語事前学習済み BERT モデルを用いた。使用するデータは、アンサーペアデータを用いて、正例データと負例データを作成した。正例データには、アンサーペアデータをそのまま使用した。負例データは、アンサーペアデータの入力バースに対して、データ内のアンサー文を、重複なしにランダムにペアとするネガティブサンプリングにて作成した。

学習には、図 4 に示すように、アンサーペアデータを用いて、入力バースを 1 文目に、アンサーを 2 文目に入力し、アンサーであるかの判別をおこなった。ハイパーパラメータは表 2 に示す条件でおこなった。

## 4.3 バース内のラップ選択

バース内のラップ選択では、アンサーを考慮したラップ選択と同様に、BERT を用いた隣接文予測をおこなった。使用するデータは、ラップバトルコーパス内のバース内の文をペアにしたデータである。バース内のあるの文に対して、その次の文をペアにする。これをバースの始めからおこなうことで、隣接する文のペアができる。このデータを正例データとし、負例データには、ある文に対して、ランダムに隣接しない文をペアとした。本データを用いて、BERT の隣接文予測をおこなった。ハイパーパラメータと用いた BERT モデルはアンサーを考慮したラップ選択と同条件である。

## 5 結果と考察

本節では、まず、順方向での生成結果と、逆方向での生成結果の違いを比較する。次に、ライムを考慮した逆向き生成の生成結果を示す。その後、全体の生成結果を示す。文の生成は、すべて Greedy サーチでおこない、終端記号が生成された時点で生成を終了した。逆向き生成においては、生成したトークン列を反転させることで、最終的な出力を得た。生成文に、同一トークンが繰り返し生成される文は削除するフィルタリングをおこなった。

### 5.1 順方向生成と逆方向生成の比較

順方向と逆方向でファインチューニングしたモデルの生成結果を表 3 と表 4 に示す。表の左側に記載している入力文が、生成モデルに入力した文であり、各モデル名が書かれた列がそれ

ぞれの生成結果を記載している。

まず、BERT2BERT と RoBERTa2RoBERTa においては、内容は少し異なる場合があるが、順方向と逆方向において、生成文に大きな違いは見られなかった。また、内容についても、ラップ調かつラップバトルで用いられるような、「勝てねえ」や「優勝」といった語が生成できていることがわかる。このことから、逆向き生成のトークンを反転させた学習は、うまく学習できているといえる。

しかし、GPT-2 では、逆向き生成において、生成文が非常に短くなる傾向が見られた。順方向の生成では、入力文の続きのような長いラップ調の文を生成しているが、逆向き生成では、非常に短い文が生成されている。GPT-2 は、これまでに生成した文の続きを生成する単方向に長けているモデルであるため、このような、逆方向の生成では、事前得られる情報が少ないため、生成がうまくおこなえなくなった可能性がある。

### 5.2 ライムを考慮した逆向き生成の結果

前節の結果から、RoBERTa2RoBERTa と BERT2BERT に大きな違いは見られなかったため、BERT2BERT と GPT-2 の結果を示す。各モデルの入力は、ラップバトルコーパスからランダムに取得した実際のバースを用いた。今回、ライムは「AIOU」で一致する語を指定した。指定した語は「帝王、概要、才能、大脳、解法、財宝、太陽、対応、内容、栄養」の 10 語である。このライムを指定した生成した結果を表 5 に示す。

まず、BERT2BERT での生成結果では、指定したライムを末尾に含んでいることが確認できる。それぞれのラップには、ラップバトルに用いられる、「俺」や「決勝」や「上がってきた」のような表現が含まれている。また、ライムに対して自然な文が生成できていることがわかる。たとえば、「俺が持ってきたお前の財宝」や「まるでギャク漫画みたいな内容」など、末尾のライムと文脈が自然である。最後に、GPT-2 では、末尾の語のみを生成してしまっている場合や、かなり短い文長での生成が多いことがわかる。これは前節でも述べたように、モデル構造が逆向きの生成に適していないことが原因と考えられる。

### 5.3 アンサーを考慮したラップ選択

各選択手法における結果を示す。まず、アンサー文ランキングでは、アンサーペアデータから、テストデータ 146 件に対して分類をおこなった。結果として、正解率が 0.746 であった。分類結果の混同行列を図 5 に示す。おおむねアンサーを分類できていることがわかる。まれに、関連ありと判断すべき文を関

表 3 順方向での生成の比較

入力文	BERT2BERT	RoBERTa2RoBERTa	GPT-2
噂のように流れるフロー	俺はこの場では終わらない	俺は俺のラップでラップしてる	で俺は今日もマイクを握ってる
人を否定して自分の音楽あげようとするなよ	お前のラップは俺の方がいいんだよ	俺は俺のやり方でやってる	俺はお前に負けないくらいのラップをしたいだけなんだ
お前自己紹介で初めて会った時	俺はこの場で優勝した	お前のラップはどうだ	よりも俺は成長したってことさっきから聞いてるけどお前のラップは相変わらずダサイよなあ

表 4 逆方向での生成の比較

入力文	BERT2BERT	RoBERTa2RoBERTa	GPT-2
噂のように流れるフロー	俺のラップでかましてんだよ	俺はそんな感じだぜ	俺のライムと雨
人を否定して自分の音楽あげようとするなよ	自分でラップしてんだよ	俺に勝てねえとか言ってんじゃねえんだよ	今日も雨の日
お前自己紹介で初めて会った時	お前のことを気にしてんだよ	って思ってるのか思ってるじゃねえんだよ	今日も雨の日

表 5 逆向き生成の各手法の比較：AIOU で一致するライムを考慮

入力バース	BERT2BERT	GPT-2
よう忘れてねーぜお前のベストエイト	俺はここまで上がってきたぜ平成の <b>帝王</b>	俺は俺の <b>帝王</b>
去年お前に言われたこと	それが俺の人生の <b>概要</b>	<b>概要</b>
俺のギャラが一勝数千円	俺はマイクでラップしてるだけの <b>才能</b>	俺のラップの <b>才能</b>
だけど1年間頑張ってるこらえて	去年の決勝に上がってきたってのは俺の <b>大脳</b>	してんのは俺の <b>大脳</b>
初めてゲットしたぜ万札のピン	お前の一刀流の <b>解法</b>	してんのは俺の <b>解法</b>
だからお前をぶち殺す	俺が持ってきたお前の <b>財宝</b>	王冠の <b>財宝</b>
お前を2年間応援し続けた	俺はお前をむき出しの <b>太陽</b>	俺は <b>太陽</b>
お前が負ける姿も見た	俺は俺の方への <b>対応</b>	してんのは俺の <b>対応</b>
それを男気が俺が次はしょって	まるでギャグ漫画みたいな <b>内容</b>	<b>内容</b>
東京に行って新兵だ	それが俺の <b>栄養と栄養</b>	お前の <b>栄養</b>

表 6 ライムとアンサーを考慮した生成結果：AUEI で一致するライムを考慮

入力バース	BERT2BERT
よう忘れてねーぜお前のベストエイト	俺は自分のことを大事にしてんだぜ俺は東京の <b>各停</b>
去年お前に言われたこと	俺はここでやってんだぜ俺は東京の <b>各駅</b>
俺のギャラが一勝数千円	俺は自分でやってんだぜ俺の <b>戦略爆撃</b>
だけど1年間頑張ってるこらえて	お前と二人でやってやってんだよ俺は日本からの <b>留学生</b>
初めてゲットしたぜ万札のピン	お前は何言ってんだよお前の <b>学歴</b>
だからお前をぶち殺す	お前の調子はどうなんだよお前は <b>小学生</b>
お前を2年間応援し続けた	俺がここで上がってきたことを大事にしてんだぜ <b>発令</b>
お前が負ける姿も見た	分かってないって言ってんのか俺は同系の <b>学生</b>
それを男気が俺が次はしょって	お前の卒業じゃなくてお前の <b>卒業写真撮影</b>
東京に行って新兵だ	お前のことを馬鹿にしてからの <b>革命</b>
	分かってないってことが分かってないってのは <b>脆弱性</b>

連なしと分類していることがわかる。

次に、隣接文ランキングでは、ラップバトルコーパスから、テストデータ 1076 件に対して分類をおこなった。結果として、正解率は 0.6 であった。分類結果の混同行列を図 6 に示す。結果として、関連なしに分類される傾向にあることがわかる。これは元々のラップバトルのバース自体が文脈を考慮しにくい

データであることが原因と考えられる。

#### 5.4 ライムとアンサーを考慮したバース生成

ライムとアンサーを考慮したバース生成結果を表 6 に示す。生成するライムは、「AUEI」を対象として生成した。考慮するアンサーは、入力バースの最後の文である「東京に行って新兵

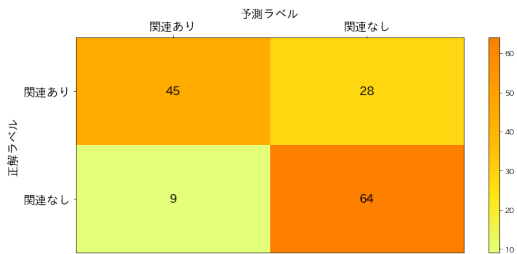


図5 アンサー文ランキングの混同行列

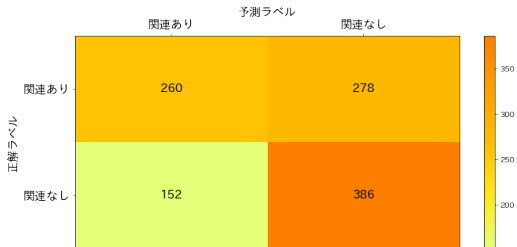


図6 隣接文ランキングの混同行列

だ」とした。まず、アンサーを考慮した選択では、最初の文に、「俺は自分のことを大事にしたい俺は東京の各停」という、相手の「東京」という発言を考慮したラップが選択できている。また、バース内のすべてのラップにおいても、「AUEI」で一致するライムを考慮した自然なラップが生成できている。以上のように、本提案手法では、ライムとアンサーを考慮したバース生成がおこなえることを示した。

## 6 まとめと今後の課題

本論文では、逆向き生成を利用したライムとアンサーを考慮したバースの生成手法について述べた。提案手法では、生成と選択の2つステップにて、バースの生成をおこなった。生成のステップでは、ライムを考慮する過程で逆向きに生成する手法を提案し、その順方向との影響の比較についても言及した。選択のステップでは、BERTモデルを隣接文予測にて学習し、アンサーとバースを構成する文の選択をおこなった。ライムを考慮した逆方向生成では、ライムを含むラップ文を生成することを示した。アンサーの選択においても、高い正解率で、アンサーを選択できることが示された。今後、人手によるバースの評価をおこなう。評価者はラップバトルを3年以上にわたって閲覧している評価者を対象に複数人で評価する。本研究では、ラップバトルの評価観点から、文の自然さ、ラップらしさ、ライム、アンサーの4つの指標に基づいて評価をおこなう予定である。評価にはラップバトルコーパスのテストデータからランダムにバースを取得し、バースに対する生成したバースを評価する。生成したバースと実際の返答であるバースを混ぜた人手評価をおこなう予定である。

## 謝辞

本研究は JSPS 科学研究費助成事業 JP21H03775, JP21H03774, JP21H03554, ならびに、2021 年度国立情報

学研究所公募型共同研究 (21S1001) の助成を受けたものです。ここに記して謝意を表します。

## 文献

- [1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL'19*, pp. 4171–4186, 2019.
- [2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR'15*, 2015.
- [3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [4] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of SIGKDD'16*, pp. 195–204, 2016.
- [5] Enrique Manjavacas, Mike Kestemont, and Folgert Karsdorp. Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *Proceedings of INLG'19*, pp. 301–310, 2019.
- [6] Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *Proceedings of INLG'20*, pp. 360–373, 2020.
- [7] Peter Potash, Alexey Romanov, and Anna Rumshisky. Ghostwriter: Using an lstm for automatic rap lyric generation. In *Proceedings of EMNLP'15*, pp. 1919–1924, 2015.
- [8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI.
- [9] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. In *Proceedings of TACL'20*, pp. 264–280, 2020.
- [10] Richard Savery, Lisa Zahray, and Gil Weinberg. Shimon the rapper: A real-time system for human-robot interactive rap battles. *arXiv preprint arXiv:2009.09234*, 2020.
- [11] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.
- [12] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceedings of NIPS'14*, 2014.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Proceedings of NIPS'17*, pp. 6000–6010, 2017.
- [14] Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*, 2019.
- [15] Dekai Wu and Karteek Addanki. Learning to rap battle with bilingual recursive neural networks. In *Proceedings of IJCAI'15*, pp. 2524–2530, 2015.
- [16] Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. Deep-rapper: Neural rap generation with rhyme and rhythm modeling. *arXiv preprint arXiv:2107.01875*, 2021.
- [17] 稲葉通将. おーぶん 2ちゃんねる対話コーパスを用いた用例ベース対話システム. 第 87 回言語・音声理解と対話処理研究会 人工知能学会研究会資料 SIG-SLUD-B902-33, pp. 129–132, 2019.
- [18] 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. 第 13 回データ工学と情報マネジメントに関するフォーラム'20, 2020.