Vol.2020-ICS-198 No.5
2020/3/7

深層学習を用いた言語モデルによる俳句生成におけるトークン単位選択

横山 想一郎[1,a)]　高橋 遼[2,b)]　山下 倫央[1,c)]　川村 秀憲[1,d)]

概要：近年, 深層学習が芸術分野へ応用され, 文章生成でも大きな成果を挙げつつある．本稿では Long-Short Term Memory(LSTM) を用いた言語モデルを中心として，俳句生成を行うシステムを構築することを目指す．言語モデルの構築に際して，トークン単位の違いや韻律の区切り方の違いが推定精度に及ぼす影響を検証した．

Tokenization Unit Slection on Haiku Generation using Neural Language Model

1. はじめに

近年，深層学習が芸術分野へ応用され，文章生成でも大きな成果を上げつつある．文章生成の試みとしては人狼ゲームのログを用いた小説の生成 [1] や RNN による漢詩の生成 [2] などがある．

文章による芸術の一分野として俳句がある．俳句は日本で古くから親しまれている伝統芸能であり，一般的に 5・7・5 の計 17 音から構成され，季節を示す「季語」や余韻を生むとされる「切れ字」を含む．

俳句はある程度定式化されたルールの下で作られるため，各種評価を定量的に行うことが比較的容易である．また，古くから日本で親しまれているために，学習データを豊富に用意することが可能である．本研究では，深層学習の芸術分野への応用例としてこの俳句を選択し，その可用性を検討する．

コンピュータによる俳句生成は以前から試みられている．近年は Wu ら [3] の研究のように，深層学習を用いた言語モデルによる生成が主である．ここで言語モデルの評価は主にパープレキシティを用いて行われているが，語彙数が異なると単純に数値を比較できないなどの問題を抱えている．また，言語モデルを構築する際に，トークン単位の違いが俳句生成に対してどのような影響を与えるか検証された例は少ない．

本稿では，言語モデルに俳句データを学習させる際，トークン単位の選択が俳句生成に対してどのような影響を与えるかを検証する．また，韻律の区切れ目を明示して学習を行うことが，生成俳句の質を向上させることに寄与するかについての検証を行う．

2. 関連研究

伊藤ら [4] は俳句生成のアプローチを，品詞や共起情報と型を用いたトップダウン的な手法と深層学習を用いたボトムアップ的な手法の 2 種類に分類し，両者を融合させた多重的アプローチについて検討を行った．トップダウン的な生成手法としては，俳句データにおける品詞の出現パターンから品詞の遷移列を抽出し，該当する品詞の単語をランダムに選択することで俳句を生成する手法と，共起関係のある単語を選択することで俳句を生成する手法を提案した．このとき，利用する単語や共起頻度については，小方らによる統合物語生成システム [5] の名詞概念辞書と動詞概念辞書から取得した．ボトムアップ的な手法としては，深層学習を用いた言語モデルを俳句データにより学習し，得られた言語モデルを用いて俳句を生成する手法を提案した．言語モデルの入出力単位としては，単語レベルや文字レベルなど複数の方法が試みられているが，学習に用いた

[1] 北海道大学大学院情報科学研究院
〒 060-0814 札幌市北区北１４条西９丁目
[2] 北海道大学工学部
〒 060-0814 札幌市北区北１４条西９丁目
[a)] yokoyama@complex.ist.hokudai.ac.jp
[b)] r.takahashi@complex.ist.hokudai.ac.jp
[c)] tomohisa@complex.ist.hokudai.ac.jp
[d)] kawamura@complex.ist.hokudai.ac.jp

俳句データは 62 句と言語モデルの学習データとしては小規模であり，学習の結果として得られる言語モデルは学習元の俳句データとほぼ同一となることが報告されている．

太田ら [6] は，深層ニューラルネットワークを用いて，単語列を入力として関連するキーワードを含む俳句を出力する俳句生成器を提案した．俳句を出力する際には再帰層に LSTM セルを使用した注意機構付きの系列変換モデルをベースとしたモデルを用いている．俳句のデータセットとしては，いくつかの web サイトから 31,509 句の学習データと 3,500 句のテストデータが作成された．俳句を出力する系列変換モデルに対して，単語の拍数を表す「拍数素性」を導入し，正しい拍数の俳句を生成する割合が高くなることが検証された．また，入力単語列が関連する季節と同じ季節についての俳句を生成するように「季節素性」を導入し，入力単語列と同じ季節に関する俳句を出力する割合が高くなることが検証された．

筆者ら [7] は，俳句を学習した言語モデルが俳句の特徴を含む多様な文字列を生成可能であり，大半は俳句としての解釈が難しいものの，一部には俳句として十分に成立するものが含まれていることに着目し，言語モデルの出力に対して俳句としての質を評価することが可能なモデルの獲得を通じて，俳句の自動生成を行う枠組みの実現を目標とした研究を行ってきた．

俳句データから任意の 2 つの形態素を選択し交換したデータを作成することで，俳句データに存在しない単語列を生成し，こうしたデータと俳句データとの分類を Bidirectional LSTM (BLSTM) により学習した．言語モデルから出力された単語列を獲得されたモデルに入力し，俳句データの確率が高いと推定された文字列を観察することで，獲得されたモデルの俳句評価モデルとしての利用可能性を検証した．また，俳句を学習する言語モデルの構造を変更し，出力される文字列に対して俳句としての拍数の正しさなどを定量指標として評価することで，俳句の学習に適した言語モデルの構造を検証した．

3. 俳句データセット

俳句は世界最短の定型詩とされている．その源流は近世に発展した俳諧にあると言われ，現在でも多くの人々に親しまれている伝統芸能である．公益社団法人日本伝統俳句協会[*1]の定義によれば，俳句とは以下のルールを満たすものである．

- 5・7・5 の 17 文字 (音) で構成される
- 季節の言葉 (季題) を含む

17 音のうち最初の 5 音は「上五」，次の 7 音は「中七」，最後の 5 音は「下五」と呼ばれる．「季題」は「季語」と呼ばれることもあり，それぞれの季語は特有の背景的な意味を有する．この背景的な意味は「本意・本情」と呼ばれ，歳時記などを通じて俳句を詠む人々の間で共有されている．また，「切れ」を生み出すために「や」「かな」「けり」などの「切れ字」を含んでいる句が多い．この切れは句に空間をもたらし、詠嘆や感動をより深くさせる効果があり，17 音という非常に短い音数を最大限活用するために重要な要素となっている．

非常に有名な句の一つに「古池や蛙飛び込む水の音 (松尾芭蕉)」という句がある．この句は「蛙」という春の季語を含み，「や」が切れ字である．一般的には「古池に蛙が飛びこんで水の音がした」という内容の句であると解釈されているが，鑑賞する人によって別の解釈がされることもある．句の解釈が一意に定まらないことは，俳句特有の難しさであり面白さでもある．

3.1 俳句の種類

5・7・5 のリズムを基本とする俳句を「定形俳句」といい，5・7・5 のリズムにとらわれない俳句を「自由律俳句」という．また季語を含む俳句は「有季俳句」と呼ばれ，季語を含まない俳句は「無季俳句」と呼ばれる．本研究では，俳句の中でも特に「定形俳句」と「有季俳句」の双方の条件を満たす「有季定型俳句」のみを取り扱う．5・7・5 のリズムを意識しつつも，音数がこれに一致しない句は「字足らず」や「字余り」と呼ばれるが，本研究では厳密に 5・7・5 に従う俳句だけを有季定形俳句として定義する．

3.2 本稿で対象とする俳句

本稿では，インターネット上の俳句収集サイトから得られた現代俳句の俳句データの中から，17 音により構成され，句またがりでなく，季語を 1 つのみ持つ 157,340 句を対象とする．有季定形俳句が満たすことが望ましい項目として，機械的に処理可能な下記の項目を用いる．

3.2.1 17 音により構成されること

有季定型俳句は基本的に 17 音で構成される．形態素解析器の MeCab[8] を利用し，俳句データを形態素解析したのち，形態素の音数を合計し 17 音であることを俳句の条件とする．ただし，辞書は新語に強いとされる mecab-ipadic-neologd[*2]を使用した．

3.2.2 句またがりでないこと

上五，中七，下五の複数に渡って一つの単語がまたがる時，これを「句またがり (破調)」と言う．句またがりは効果的に使用すれば句に独特の効果をもたらすことも多い．しかしここでは，単語がまたがる場合は俳句のリズムを学習できていないと考え除外することにする．俳句を形態素分割したとき，5 音目と 12 音目のどちらかに単語がまたがるものを句またがりと判定する．

[*1] http://haiku.jp/

[*2] https://github.com/neologd/mecab-ipadic-neologd

3.2.3 未知語を持たないこと

形態素解析器が未知語と判定した単語を含むものは，俳句でないものとみなす．

3.2.4 季語を 1 つのみ持つこと

インターネット上の季語データベースからスクレイピングして収集した 8,642 語の季語データを用いて，季語数が 1 以外の句を除外する．同じ季語を複数回含む場合は，区別してカウントする．例えば「八重桜見れば彼方も八重桜 (菖蒲あや)」という俳句は「八重桜」という 1 種類の季語しか使用していないが，重複して 2 度使用しているので季語数は 2 とカウントする．

3.2.5 切れ字を 2 つ以上持たないこと

切れ字数が 1 以下でない句を除外する．なお上五，中七，下五の末尾が以下の条件に当てはまる場合その単語を切れ字と判定する．

- 助詞の「や」
- 助詞「か」と助詞の「な」の連続
- 助動詞の「けり」

4. 俳句を学習する言語モデル

言語モデルは LSTM で構成されている．学習段階と俳句生成段階で動作が異なるため，分けて説明をする．

4.1 俳句学習

図 1 に学習のイメージを示す．まず，トークン単位に従って俳句を分割する．それぞれのトークンに対して出現順の若い方から順番に ID を与え，それぞれのトークンに割り当てられた ID を基に，俳句を ID の列に変換する．次に，ID の列を先頭から順番に LSTM に入力していく．この時 ID は 1-of-k 符号化されて入力され，LSTM は次に来るトークンの確率列を出力する．ここで出力されるベクトルの i 番目の要素は，ID が i のトークンが次に来る確率である．全ての ID を入力した段階で誤差を計算し学習を進めていく．パラメータ設定は表 1 の通りである．実装は TensorFlow[*3]で行った．

表 1 言語モデルを学習する際のパラメータ設定

パラメータ名	設定値
LSTM 層数	2
LSTM ユニット数	128
最適化手法	Adam[9]
学習率	0.001
エポック数	1600
バッチサイズ	2048
目的関数	クロスエントロピー

[*3] https://www.tensorflow.org/

4.2 俳句生成

学習済みの言語モデルを用いて俳句をする手順について説明する．生成のイメージは図 2 に示す通りである．まず，初期入力として BOS を入力する．次に来るトークンの確率を計算し，その確率に基づいて次のトークンをルーレット選択する．EOS が出現するまでこれを継続することで，俳句を生成する．

5. 実験

本稿では，言語モデルに俳句データを学習させる際，トークン単位の選択および韻律の区切れの明示が生成する文字列の性質に与える影響を検証する．

先述の通り，インターネットから収集された，17 音により構成され，句またがりでなく，季語を 1 つのみ持つ 157,340 句を学習データとする．獲得された言語モデルが出力する文字列について，先述の本稿における俳句としての条件を満たす割合を検証する．

また，学習設定によっては，言語モデルが学習データと全く同じ俳句，あるいは酷似した俳句を生成する可能性がある．すべての学習データとの編集距離 [10] を計算し，その最小値を取得することでその俳句が学習データに酷似していないかを判定する．本稿では編集距離の最小値が 5 以下の俳句を類似句と定義してその割合を示す．

5.1 トークン単位の違いによる言語モデルの比較

言語モデルを取り扱う際，文章の最小単位をどのように定義するかは非常に重要な問題である．この最小単位のことをトークンという．この実験では言語モデルにおけるトークン単位として，文字，単語および Sentencepiece[11] の 3 種を比較し，トークン単位の選択が俳句生成に与える影響を調べる．

Sentencepiece は高頻度で出現する文字列をひとまとまりとする分かち書き手法であり，語彙数を効率的に削減することができる．Sentencepiece モデルの学習については，すべての文字を被覆し，語彙数が 8，000 となるように設定した．なお各トークン単位の下での語彙数は表 2 に示す通りである．

表 2 各トークン単位の下での語彙数

トークン単位	語彙数
文字	4,414
単語	53,321
Sentencepiece	8,000

トークン単位以外の設定は表 1 と同じものを用いた．この設定のもと，各トークン単位を用いて 4 回ずつモデルを学習させた．それぞれのモデルから 1 万句ずつ俳句を生成させ，各トークン単位で 4 万句ずつを用意した．これらの出力俳句を定量的な指標の元で比較することで，トークン

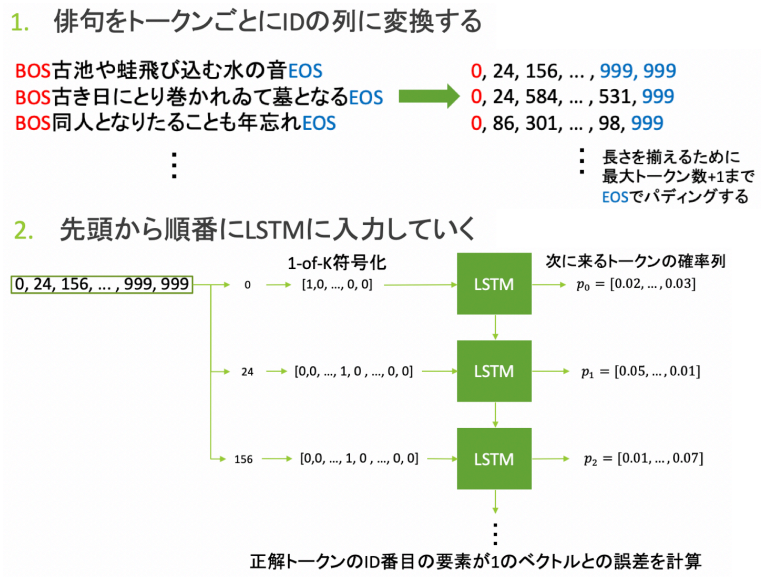

図 1 言語モデルの学習イメージ

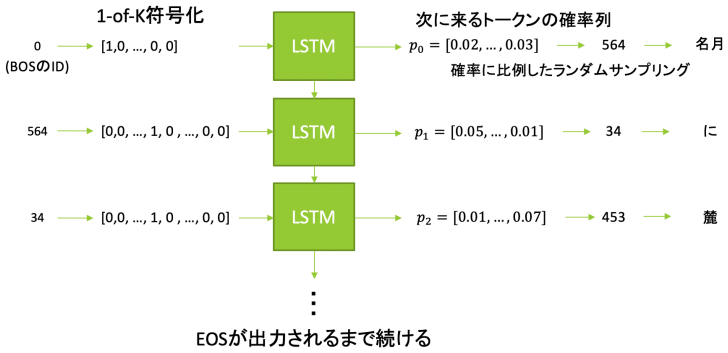

図 2 言語モデルによる俳句の生成イメージ

単位選択の影響を検証した．

5.1.1 各種条件を満たす割合の比較

まず，それぞれのトークン単位の下で生成された俳句の各種の条件を満たす割合を調べた．結果は表 3 に示す通りである．最も右の列は全ての条件を満たす俳句の割合を示す．

トークン単位として単語を選択した言語モデルが，学習データに最も近い性質の俳句を生成できていることがわかる．特に出力された俳句が 17 音で構成される割合と句またがりでない割合が高いことから，俳句のリズムをうまく学習できていると考えられる．また，季語数が 1 である俳句の割合も高く俳句の特徴をよく捉えている．未知語を含まない句の割合が学習データより高いことも特徴的である．一方，類似句の割合は 3 種の中でも最低であり，学習データに似ている俳句を生成してしまっていることがわかる．

Sentencepiece と文字を比較すると，17 音の句の割合や句またがりでない句の割合は文字の方が高いが，未知語を含まない句や季語を 1 つ含む句の割合は Sentencepiece の方が高い．また，類似句でない句の割合は文字の方が少し高い．

切れ字に関しては，どのトークン単位を用いても切れ字数が 1 以下の俳句を高い割合で生成できていることがわかる．

全条件を満たす俳句の割合は，トークン単位として単語を選択した場合が最も高く，定量的な観点から最も質の高い俳句を生成できているといえる．

5.1.2 最小編集距離の分布の比較

次に，生成された俳句がどの程度学習データに類似しているかを検証するために，生成俳句と学習データの編集距離を計算しその分布を調べた．結果は図 3 に示す通りである．

トークン単位として文字や Sentencepiece を用いたときは，学習データと異なるオリジナルな俳句を多く生成できている一方で，単語単位で学習を行ったときは学習データと全く同じ俳句を生成してしまっている割合が高いことがわかる．最頻値はどのトークン単位でも 8 である．

単語単位での学習を行った場合，学習データの性質をよく反映した俳句を生成できるが，学習データと全く同じ俳

表 3 各種条件を満たす割合

トークン単位	17 音	句またがり	未知語	季語	切れ字	類似句	全通過
文字	70.9%	81.0%	87.1%	83.8%	99.3%	99.2%	50.2%
単語	86.1%	91.4%	92.5%	92.1%	99.2%	77.9%	54.0%
Sentencepiece	68.7%	80.2%	87.7%	87.6%	99.4%	98.3%	51.2%
学習データ	100.0%	100.0%	91.9%	100.0%	99.9%	-	-

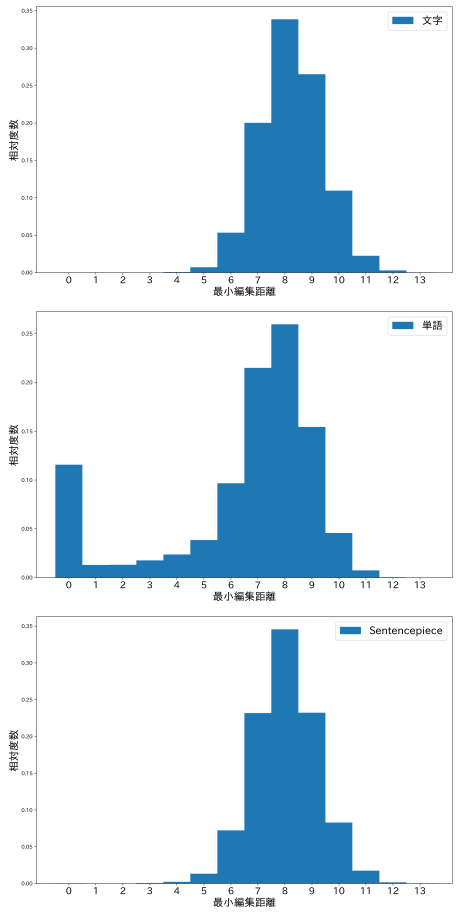

図 3 最小編集距離の分布

句を生成する確率も高まることがわかる．

5.2 韻律の区切れの明示による言語モデルの比較

5・7・5 というリズムは，俳句を俳句たらしめる非常に重要な要素である．一方で学習データとして準備した俳句データは，上五，中七，下五が結合した文字列であり，5・7・5 の区切れ目の情報を保持しない．ゆえに学習データをそのまま用いると，言語モデルはこのリズムを十分に学習できていない可能性がある．そこで，この韻律の区切れ目を明示して学習を行うことで，5・7・5 のリズムを守った俳句の生成割合が高まると考えた．よってこの実験では，俳句データの韻律の区切れ目に記号を挿入するという前処理を行い，言語モデルの学習を行うことを試みた．

上五，中七の区切れ目と，中七，下五の区切れ目は性質が異なると考えられる．よって，この 2 種類の区切れ目を区別して韻律の区切れを明示する前処理を行ったデータも用意した．

表 4 に示す通り，3 種類の学習データを用意し，表 1 の設定のもとで言語モデルを学習させた．本実験ではトークン単位に文字単位を選択した．

各前処理を施した学習データを用いて 4 回ずつモデルを学習させた．それぞれのモデルから 1 万句ずつ俳句を生成し，各前処理の方法について 4 万句の生成データを用意した．

表 4 学習データに対する前処理の種類

前処理の種類	学習データ例
前処理なし	古池や蛙飛び込む水の音
区切れ目を区別しない	古池や｜蛙飛び込む｜水の音
区切れ目を区別する	古池や｜蛙飛び込む$水の音

5.2.1 各種条件を満たす割合の比較

各前処理を施した学習データを用いて学習した言語モデルが出力した俳句の，各種の条件を満たす割合を調べた．結果は表 5 に示す通りである．最も右の列は全ての条件を満たす俳句の割合を示す．

2 種類の区切れ目を区別しない前処理を行った学習データを用いて学習させた言語モデルが，最も学習データに近い性質の俳句を生成できていることがわかる．前処理をしない場合と比較して，前処理を施した場合は出力が 17 音で構成される割合と句またがりでない割合が上昇している．これは，言語モデルが俳句のリズムを捉えられている結果

表 5　各種条件を満たす割合の通過率

前処理の種類	17 音	句またがり	未知語	季語	切れ字	類似句	全通過
前処理なし	70.9%	81.0%	87.1%	83.8%	99.3%	99.2%	50.2%
区切れ目を区別しない	74.8%	83.9%	87.5%	86.4%	99.4%	99.1%	55.9%
区切れ目を区別する	74.6%	83.7%	87.6%	85.2%	99.2%	99.1%	55.2%
学習データ	100.0%	100.0%	91.9%	100.0%	99.9%	-	-

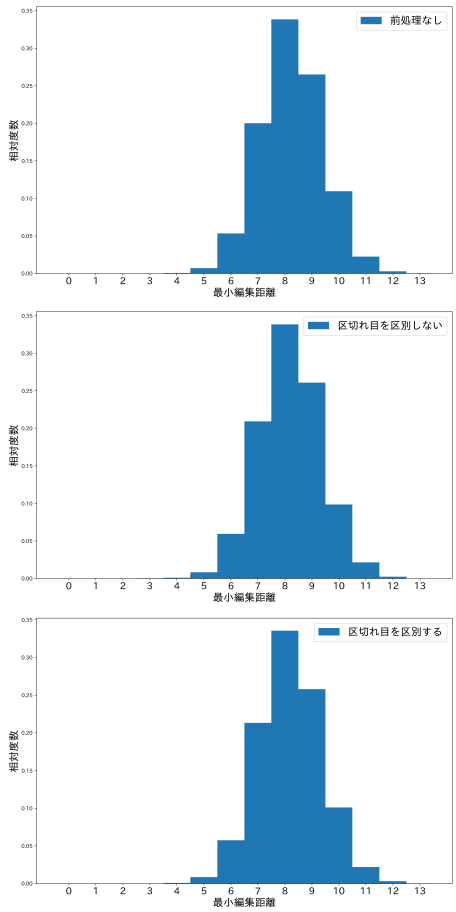

図 4　最小編集距離の分布

と捉えることができる．また，季語を 1 つのみ含む割合も比較的大きく上昇していることがわかる．

類似句の割合は全てのモデルでほとんど同じ値となっており，前処理の導入によって言語モデルが効果的に俳句の特徴を捉えられるようになったと考えられる．

5.2.2　最小編集距離の分布の比較

生成された俳句がどの程度学習データに類似しているかを検証するために，生成俳句と学習データの編集距離を計算しその分布を調べた．結果は図 4 に示す通りである．

すべてのモデルにおいて，ほぼ同一の分布となっている．このことから，前処理の導入により学習データに類似することなく，俳句のルールをより学習できていると言うことができる．

6.　おわりに

本研究では現代俳句を用いて，様々な条件下で言語モデルを学習させ俳句を生成した．トークン単位として単語を用いた言語モデルが学習データに最も近い性質の俳句を出力した一方で，学習データと全く同じ文字列を出力する確率が高いことも示された．また，韻律の区切れを明示して言語モデルを学習させることで，より俳句のルールを捉えた俳句を生成できることが確かめられた．

謝辞　本研究成果の一部は，北海道大学情報基盤センターの人工知能対応先進的計算機システムを利用して得られたものです．

参考文献

[1] 松山諒平，佐藤理史，松崎拓也：人狼ログからの小説の自動生成，言語処理学会第 23 回年次大会発表論文集，pp. 32–35 (2017).
[2] Zhang, X. and Lapata, M.: Chinese Poetry Generation with Recurrent Neural Networks, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Association for Computational Linguistics, pp. 670–680 (online), DOI: 10.3115/v1/D14-1074 (2014).
[3] Xian-chao, W., Momo, K., Kazushige, I. and Zhan, C.: Haiku Generation Using Deep Neural Networks (2017).
[4] 伊藤拓哉，五十嵐広太，　小方孝：俳句生成への多重的アプローチの考察，人工知能学会全国大会論文集，Vol. JSAI2018, pp. 3B1OS22a04–3B1OS22a04 (2018).
[5] Ogata, T.: Computational and cognitive approaches to narratology from the perspective of narrative generation, *Computational and cognitive approaches to narratology*, IGI Global, pp. 1–74 (2016).
[6] 太田瑶子，進藤裕之，松本裕治：深層学習を用いた俳句の自動生成，技術報告 1，奈良先端科学技術大学院大学，奈良先端科学技術大学院大学，奈良先端科学技術大学院大学 (2018).
[7] 横山想一郎，米田航紀，山下倫央，川村秀憲：俳句生成を目的とした言語モデルに対する Attention 機構の導入，技術報告 3，北海道大学大学院情報科学研究院，北海道大学大学院情報科学研究科，北海道大学大学院情報科学研究院，北海道大学大学院情報科学研究院 (2019).
[8] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, Association for Computational Linguistics, pp. 230–237 (online), available from ⟨https://www.aclweb.org/anthology/W04-3230⟩ (2004).
[9] Kingma, D. P. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).
[10] LEVENSHTEIN, V. I.: Binary codes capable of correcting deletions, insertions and reversals, *Soviet Physics Doklady*, Vol. 10, pp. 707–710 (online), available from ⟨https://ci.nii.ac.jp/naid/10020212767/⟩ (1966).

[11] Kudo, T. and Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium, Association for Computational Linguistics, pp. 66–71 (online), DOI: 10.18653/v1/D18-2012 (2018).