Vol.2018-NL-235 No.1
Vol.2018-SLP-121 No.1
2018/5/13

深層学習を用いた俳句の自動生成

太田瑶子[†1]　進藤裕之[†1]　松本裕治[†1]

概要：文学の一つとして詩がある．詩は言葉の表面的な意味だけでなく，言葉が持つ美学的・喚起的な性質を用いて表現される．詩は短い文字列であっても，詩として表現する事で，言葉の持つ奥深さによってその場の雰囲気を封じ込めることが出来る．しかし，実際にいざ詩を作ろうとすると，どのように始めれば良いのか難しい．そのような場合であっても，手軽に詩を作れるようにしたいと考えた．本研究では，詩の中でも有季定型俳句を選び，言葉を入力することにより俳句の自動生成を行った．本研究ではより柔軟な表現が生成できるように，深層学習を使った．また，韻律や季語のような有季定型俳句の規定を素性や制限として用いた．俳句としての体をなすような生成結果が得られた．

キーワード：俳句，自動生成，機械学習，ニューラルネットワーク

Haiku Generation with Deep Learning

Yoko Ota[†1]　Hiroyuki Shindo[†1]　Yuji Matsumoto[†1]

Abstract: Poetry expresses things of nature and feelings in human life in rhythmic linguistic form. A message is conveyed not only by the superficial meaning of words, but also by aesthetic and evocative property of words. Subtle emotions or beauty of natural scenery can be expressed in long writings, but they can also be expressed as poems. Even if poems are short, however, it can record the atmosphere of the place by the profoundness of words. Many people learn how to make poems and how to read them in the course of the compulsory curriculum course. However, it is not easy to make poems for those who are not familiar with them. Our goal in this research is to make it easier for people to write poems. In this research, we selected seasonal standard haiku in poetry and investigated methods for automatic generation of haikus. Two lines of research exist in the previous work. One is the rule-based generation of haiku and the other is a deep learning-based approach that does not take seasonal references (kigo) into account. In this research, we utilize deep learning to make more flexible generation. We also impose features to control the number of moras in the deep learning model.

Keywords: haiku, automatic generation, machine-learning, neural networks

1. はじめに

1.1 背景

文学の一つとして詩がある．詩とは自然の風物や人間生活での心の動きをリズムのある言語形式で表現したものである．それは言葉の表面的な意味だけでなく，言葉が持つ美学的・喚起的な性質を用いて表現される．その場の微妙な心情や情景を保存したいと思った際，カメラや録音機など何も持ち合わせていなくても，詩として表現すればそれらを封じ込める事が出来る．詩には字数や韻律などに規定がある定型詩と，そのような規定が特にない自由詩や散文詩がある．

しかし，詩のルールの有無に関わらず，詩は言葉の持つ奥深さを自在に扱う必要があるので，詩に慣れ親しんでいない人がいざ作ろうとしても，どのように作り始めれば良いのか分からずなかなか手を出しづらい．そのような人でも手軽に詩を作れるようにしたいと考えた．本研究では，詩の中でも日本の詩の一つである俳句を選び，その自動生成器の作成を試みた．

1.2 本稿の構成

本稿の構成について述べる．次節で，まず俳句について説明し，俳句の自動生成についての先行研究を紹介する．そして提案手法とその基本となる技術について述べる．続く節で，実験の際に用いたデータセットと，実験の結果・考察について述べる．

2. 計算機による俳句の自動生成

2.1 俳句について

俳句とは, 日本の詩であり, 詩全体が基本的に 17 拍(mora)で構成される世界で最も短い詩である．日本語における拍とは, 基本的には平仮名 1 文字が 1 拍となる. ただし,“っ”以外の小書き文字（“ゃ”や“ぇ”など）はその前の平仮名と一緒に，例えば“しゃ”で 1 拍と数えられる．俳句は 3 つのフレーズで構成され，それぞれ上五，中七，下五と呼ばれている．俳句には以下のような 3 つの特徴がある．

- 上五，中七，下五はそれぞれ 5 拍，7 拍，5 拍
- 1 俳句につき，季節を表す 1 語(季語)
- 1 俳句につき，切れや「や」「けり」のような切れ字が少なくとも 1 つ

†2 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

Vol.2018-NL-235 No.1
Vol.2018-SLP-121 No.1
2018/5/13

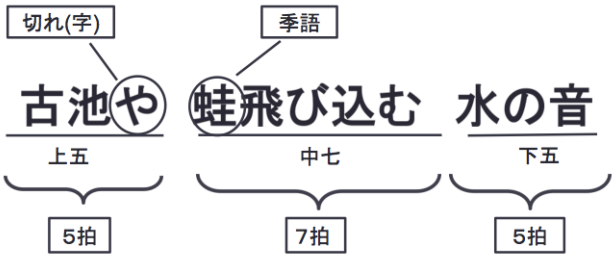

図 1 俳句の特徴(松尾芭蕉『春の日』より)

松尾芭蕉が詠んだ俳句を例として俳句の特徴を図 1 に示す．図 1 の，「古池や」が上五に該当し，「蛙飛び込む」，「水の音」がそれぞれ中七，下五に該当する．また，上五の「古池や」の「や」が切れ字であり，「蛙」が春の季語である．

上記のような特徴を含んだ，季語があり，韻律が定まっているような俳句を，有季定型俳句と呼ぶ．季節は，春，夏，秋，冬に加えて，現代では新年も含まれている．他にも季語がないが拍数は 17 拍である無季定型俳句や，上記の俳句の特徴に囚われない自由律俳句などがある．

定型俳句は，冒頭でも述べたように，基本的には全体が 17 拍になるように構成されているが，17 拍よりも少ない拍数で構成されている俳句を字足らず，17 拍よりも多い拍数で構成されている俳句を字余りという．(図 2，図 3 参照)

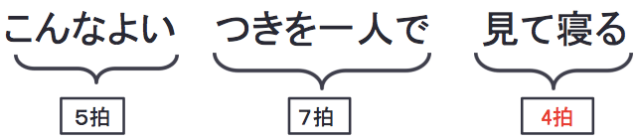

図 2 字足らずの俳句例(尾崎放哉『大空』より

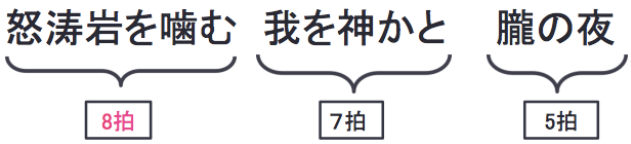

図 3 字余りの俳句例(高浜虚子『五百句』より)

字余りや字足らずの俳句にも名句と呼ばれるものが存在するが，一般的に 17 拍で出来た正しい韻律の俳句に比べて劣るとされている．

また，有季定型俳句の場合，1 つの俳句には通常 1 つの季語が入っているが，複数の季語が入っている俳句も存在する．その複数の季語が同じ季節の季語であれば，季重ねと呼ばれ，違う季節の季語であれば季違いと呼ばれる．(図 5，図 6 参照)

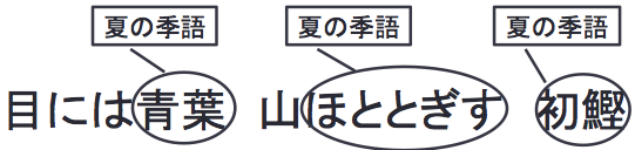

図 4 季重ねの俳句例(山口素堂 作)

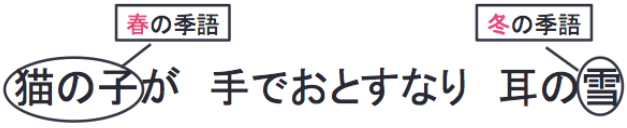

図 5 季違いの俳句例(小林一茶 作)

図 4 の俳句には夏の季語が 3 つ入っており，季重ねである．図 5 の俳句には猫の子と雪という 2 つの季語が入っており，それぞれの季節は春と冬なので季違いである．図 6 のように，隣り合っていない季節の季語でも起こりうる．

図 6 隣り合っていない季節の季違いの俳句例(松尾芭蕉『奥の細道』より

季語には強さがあり，図 5 や図 6 のような季違いの俳句では，弱い季語が無季語化し，強い季語の季節がその俳句季節を表す．

図 5 の俳句では，「猫の子」よりも「雪」が強いので，この季節は冬であり，図 6 の俳句では，「蛤」よりも「ゆく秋」の方が強いので，この俳句の季節は秋となる．

同じ季節の複数の季語が上手く生かし合っているような季重ねや，季節の違う季語がそれぞれ強い季語と弱い季語に分けられるような季違いでない限り，通常は季重ねや季違いは嫌われる．

最後に，句またがり（破調）について説明する．

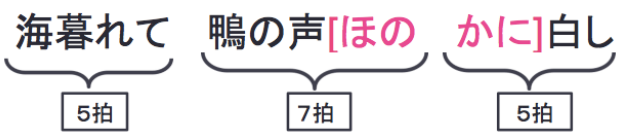

図 7 句またがりの俳句例(松尾芭蕉『野ざらし紀行』より)

句またがりとは，1 つの語が 2 つの句の切れ目をまたいでいるようなものを指す．図 7 では，「ほのかに」という語が中七と下五をまたいでいる．

本稿では，最も正統的な俳句である有季定型俳句を対象として考えていく．

2.2 先行研究

俳句の自動生成に関する先行研究として以下の 2 つを紹介する. 1 つ目はルールに基づいたもので, 2 つ目は機械学習によるものである.

2.2.1 ルールに基づいた俳句創作支援システム

土佐ら[1]の研究は，コンピューターによる俳句創作支援システム(Hitch Haiku)であり，ルールベースでテンプレートに基づいた俳句創作支援システムである．具体的には，古今東西の書を納めた図書街マップに示されたフレーズから任意のものを選択すると，そのフレーズに対応するテキストが表示される．そのテキストからユーザが俳句のベースになる語を1つ以上最大20語まで選択して俳句を生成するシステムである．ここで，俳句のベースになる語には 2 段階の優先度を付けることが出来る. 選択した語に対して，2 種類の処理を施す. 1 つ目の処理は，選んだ語の語尾や語頭の活用形に応じて，5 拍か 7 拍になるように「や」,「かな」,「けり」等の切れ字を語尾に接続する，もしくは「げに」のような副詞を選んだ語の頭に付与する処理である. 2 つ目の処理では，まず，選んだ語に関連する季語を，歳時記などの季語に関するデータベースと類語辞典の中から検索する．そして，それらを 5 拍か 7 拍に合うように組み合わせて，最も良いものを選出する，というのが 2 つ目の処理である．その後，1 つ目の処理と 2 つ目の処理で出来たフレーズを，順に 5 拍，7 拍，5 拍になるように繋いで俳句を完成させる.

2.2.2 深層学習を使った俳句自動生成

深層学習を使った俳句自動生成の研究として，Wu ら[3]の研究がある．4 種類のディープニューラルネットワークを使ってそれぞれ俳句を文字レベル学習し生成させ，パープレキシティ(perplexity)を計算して比較した論文である．ここで使われたモデルは以下の通りである.

- vanilla Recurrent Neural Network (RNN) モデル
- Long Short-Term Memory (LSTM)を使ったモデル
- Kim ら[6]の Recurrent convolutional neural networks (RCNN)モデル
- Yu ら[5]の Sequence generative adversarial networks (SeqGAN)モデル

上記のモデルをそれぞれ，手法 1，手法 2，手法 3，手法 4 とする.

手法 1 では，まず，入力層で文字をベクトルに変換し，再帰層に送られる. 時刻 t における入力層のベクトルを x_t, 再帰層の隠れ状態ベクトルを h_t とすると，次の隠れ状態ベクトル h_{t+1} はまず x_t と h_t で線形結合され，そして活性化関数を使って計算される．出力層では，1 つ前に生成した文字を考慮しながら現在生成しようとしている文字の確率を計算し，1 文字ずつ出力する.

手法 2 では，手法 1 の RNN 層を LSTM 層に変更したモデルである．これを使用して文字レベルで入出力させる. LSTM は RNN をベースとした技術である. LSTM の詳細については第 3 章で述べる．手法 1 よりも，より初めの方に入力された系列のデータを考慮しながら学習することができる.

手法 3 の文字レベルの RCNN モデルは，意味的情報と正字情報の両方を文字レベルでエンコードできる．この手法では，まず各文字をベクトルに変換する．次に，CNN[7]を使ってそのベクトルを様々なカーネルサイズで変換する. そしてベクトルは，LSTM ユニットが使用される RNN 層に送られる．最後に，次の文字を予測することを目的にして，語彙内の文字の確率を計算するために softmax 層にベクトルを送り文字を出力する.

手法 4 については, まず手法 4 の基本技術である GAN[8]について説明する．これは 2 つのネットワークを学習させる．1 つは訓練データと同じようなデータを生成しようとするネットワーク（G とする）で，もう 1 つはデータが訓練データから来たものなのか，G から来たものなのか識別するネットワーク（D とする）である．G の目標は，G が訓練データと同じようなデータを生成することである. GAN を系列データに応用し, G に対して強化学習を適用したものが SeqGAN である．基本的には GAN と同じだが，G は RNN 生成モデルをベースとしている．ある文字を入力し，次に生成すべき単語の生成確率の学習と，出力系列全体が適切な出力であるか D で判断しながら学習に反映させる.

データセットは 2 種類ある．1 つ目はチャットボット(chatbot)のクエリログ(query log)から 5 拍や 7 拍になるものを取り出し，5 拍 7 拍 5 拍になるように繋げたものを俳句として集めたもので，季語の有無は考慮していない．もう 1 つは, 幾つかの俳句の Web サイトから集めた俳句である. Web から集めた俳句を使った場合のパープレキシティは手法 2 で最も小さくなり，チャットボットから出来た俳句を使った場合は，手法 3 で最も小さくなった.

2.2.3 本研究と先行研究の違い

2.2.1 の研究と本研究の違いは，前者はルールに基づいて俳句を作っているのに対し，本研究では深層学習で俳句を生成している点である.

次に 2.2.2 の研究との違いについて述べる．この研究では 2 種類のデータセットを用いて実験を行っている．この研究の問題点の 1 つとして，データセットの 1 つがチャットボットのクエリログから作られている点である．チャットボットのクエリログには，必ずしも俳句に使われやすい表現が含まれているわけではない．また季語について考慮

していない点も挙げられる．さらに，2.2.2 の研究は俳句から俳句を生成しており，単語列から俳句を生成しようとする本研究とは異なる．

本稿では，実際の俳句のデータを単語レベルで用い，入力系列である単語列の季節を出力系列である俳句に反映できるように取り組む．

3. 提案手法

本稿の提案手法を述べる．なお，本稿では，入力系列は詠みたい俳句に関するキーワードとした．以下に例を示す．

- [入力系列] 蛙 音 池
- [出力系列] 古 池 や | 蛙 飛び 込む | 水 の 音

ここで，出力系列内の“|”は上五や中七が終了したことを表す記号として使用している．

本稿では，注意機構付きの系列変換モデルをベースとした．また，系列変換モデルの再帰層では LSTM を使用した．

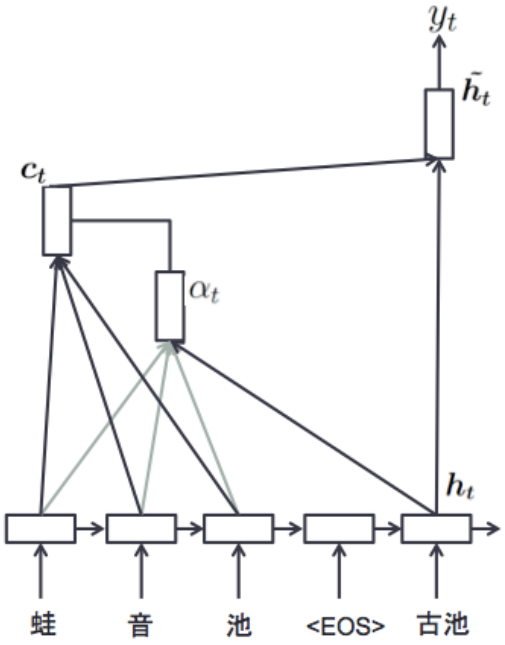

図 8 本研究の基本構造

3.1 拍数素性の導入

正しい韻律で生成されやすくなるように，系列変換モデルの Decoder 部分で入力された単語ベクトルに，拍数を表す素性ベクトルを連結させて学習を行った．定型俳句 5 拍，7 拍，5 拍に分けた場合，最大の拍数は 7 拍なので拍数の素性は 3 ビットの 2 進数で表す．

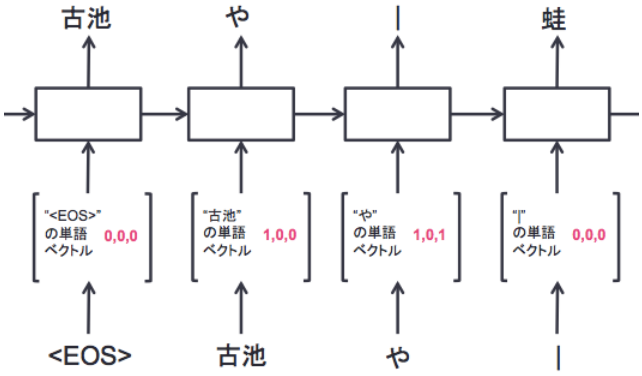

図 9 拍数素性の導入

図 9 は，Decoder の一部分を取り出して示している．また，単語ベクトルが入力されてから，予測された単語が出力されるまでの部分については，拍数素性を導入する際には変更していないので省略している．

拍数の素性ビットは，出力された単語の拍数を数えている．始めに「<EOS>」が入力される際には，図 9 のように初期値として拍数の素性ビットは[0,0,0]に設定している．その次の入力である「古池(ふるいけ)」は 4 拍なので，「古池」が入力される際の拍数素性ビットは[1,0,0]になり，1 拍である「や」を入力した際には，前の拍数素性ビットと合計した拍数が 5 なので，[1,0,1]になる．また，上五や中七の終了を表す“|”が入力されると，拍数の素性ビットは[0,0,0]になるように設定している．

最後に，拍数が 7（拍数素性ビットが[1,1,1]）を越えてしまう場合については，拍数が 7 を越えた後であっても拍数素性ビットは[1,1,1]とし続けるようにした．

3.2 季節素性の導入

入力系列と同じ季節の季語が出やすくなるように季節に関する素性を加えて学習させた．Decoder の隠れ状態ベクトル $\boldsymbol{h}$ に季語の素性を表すベクトルを連結させて素性を加えた．季語の素性は 5 つの季節を表せるように 5 ビットで表現している．

春は[0,0,0,0,1]，夏は[0,0,0,1,0]，秋は[0,0,1,0,0]，冬は[0,1,0,0,0]，新年は[1,0,0,0,0]である．また，[0,0,0,0,0]は無季節を表す．

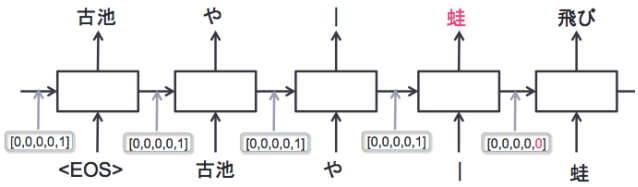

図 10 季節素性の導入

Encoder で入力された系列から季節や季語を探し，その季節の季節素性ビットを作る．図 10 では，入力系列から季節が春だと分かっているので，春の季節素性ビットを作る．

春の季語が出力されるまで, Decoder の隠れ状態ベクトル $\boldsymbol{h}$ に春の季節素性ベクトルを連結させ続ける．一度その季節の季語が生成されると，以降の隠れ状態ベクトルには無季節の季節素性を連結させる．図 10 においては，春の季語の 1 つである「蛙」が生成されているので，以降の季語素性ビットには無季語の季語素性ビットを使用する．ここで，仮に他の季節の季語が生成されてしまった場合であっても，季語素性ビットは入力系列の季節の季語が生成されるまで，無季語の季語素性ビットに変わらない．

3.3 季語制約

入力系列の季節の季語以外が出力されないように，その季節と異なる季節の季語を出力させないような制約を加えた．

Decoder で単語を予測する際に，入力系列の季節と異なる季語が選ばれそうになった場合，その単語ではなくそれ以降の確率の高い単語でかつ季語制約を満たすものを選ぶように設定した．ただし，入力系列に季節が分からないような単語列（無季節・無季語の単語列）が入力され，単語予測時に季語の単語が選ばれそうになった場合は，その季語を除くというような制約は加えていない．それは，入力系列が無季節であっても，有季定型俳句としては季語が含まれている事が大事なので，季語が含まれた系列が出力されそうになった場合はそのままそれを尊重する事にした．

4. 実験

4.1 データセットについて

本稿の実験で用いた俳句は，小林一茶や松尾芭蕉などの近代の俳句から現代の俳人が詠んだ俳句までをいくつかの Web サイトから集めたものである．([19], [20]参照)

表 1　実験で用いたデータセット

		各季節の俳句数						
	データ数	春	夏	秋	冬	新年	無	語彙サイズ
train	31,509	9,019	7,302	5,375	4,481	425	4,907	23,150
test	3,500	1,153	848	546	553	48	353	6,575

集めた俳句のうち，全体が 17 拍になっている句だけ残し，字余り・字足らず・自由律俳句は全て取り除いたので，データセットの俳句は全て定型俳句である．また，3 の提案手法で示したような入力系列と出力系列のデータの形式の関係で，句またがりの句であっても上五，中七，下五に分けられるような俳句はそのままデータセットに残し，分けられないような俳句は除いた．実験時の入力系列は，俳句から名詞を抽出して作成している．季語のデータセットについて以下の表に示す．

表 2　季語のデータセット

春	夏	秋	冬	新年
1,292	1,629	1,034	1,176	232

今回は，以下の 5 つのモデルで実験を行った．

- [Baseline] 注意機構付き系列変換モデル
- [Model1] Baseline に拍数素性を導入したモデル
- [Model2] Baseline に季節素性を導入したモデル
- [Model3] Baseline に拍数素性と季節素性を導入したモデル
- [Model4] Baseline に拍数素性と季節素性を導入し，季語制約を加えたモデル

5. 結果・考察

ここでは実験結果と考察について示す．実験結果では，拍数素性の有無による各モデルにおける拍数の変化と，季節素性と季語制約を加えた事で各モデルの出力系列の季節にどのような変化があったのか示す．なお，本研究では，上五と中七の終わりに記号“|”を入れて学習させているので，全体で 17 拍であるが 5 拍，7 拍，5 拍で分けられないような句またがりの句は生成されない．

5.1 拍数素性の有無による結果

拍数素性の有無による結果について，以下の図に示す．

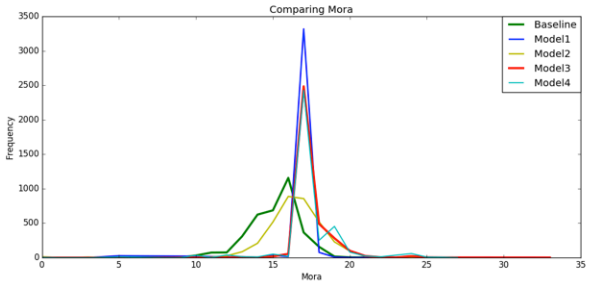

図 11　拍数の比較(全体)

図 11 は俳句全体についての拍数の頻度を表しており, 17 拍に近ければ近いほど正しい韻律で生成されているということになる．図 11 を見ると，Baseline や Model2 に比べると，Model1 や Model3，Model4 の拍数の頻度が 17 で最も多くなっており，拍数の制限をした方が，しなかった場合に比べ，正しい拍数が出力されやすいという事が分かる．

拍数素性単体を加えただけの Model1 に比べ，他の素性や制限が加えられている Model3，Model4 の，17 拍での頻度が小さくなっている理由は，選べる単語の範囲が他の素性や制限によって狭められてしまっているからだと考えられる．

Vol.2018-NL-235 No.1
Vol.2018-SLP-121 No.1
2018/5/13

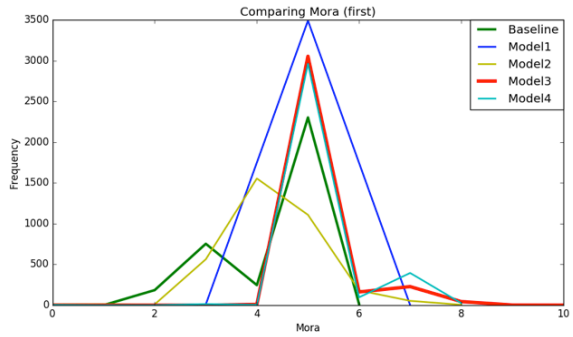

図 12　拍数の比較(上五)

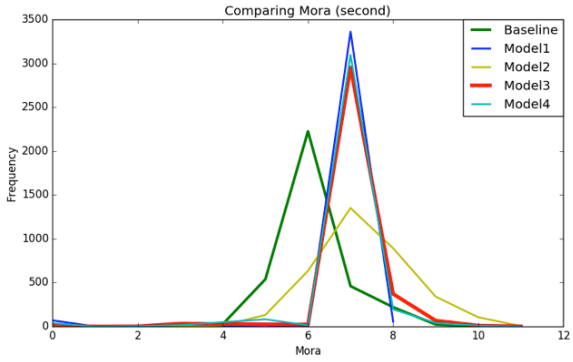

図 13　拍数の比較(中七)

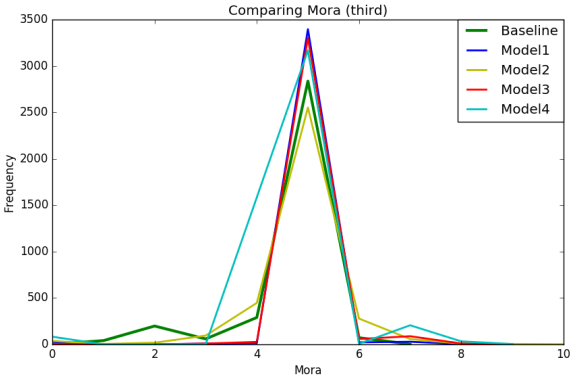

図 14　拍数の比較(下五)

図 12 から図 14 を見ると，俳句全体の拍数の場合と同じように，拍数素性を加えたモデルの方が正しい拍数で出力出来ているものが多い．7 拍が正しい拍数である中七の図 13 と，5 拍が正しい拍数である図 12，図 14 を見比べると，Baseline や Model2 が 7 拍の場合よりも 5 拍の場合の方が正しい拍数で出力されることが多いということが分かる．これは，より短いフレーズにおいては言葉を並べるだけで偶然にも正しい拍数になってしまうが，より長いフレーズになると，その偶然が起こりにくくなるからではないかと考えられる．

5.2 季節素性の導入と季語制限による結果

季節の素性導入と季語制約による結果について，それぞれのモデルについて混同行列を以下に示す．

以下の混同行列の行は Decoder から実際に出力された俳句の季節を表している．列は Encoder で入力された入力系列の季節を表している．混同行列内の各要素の値は，俳句の句数を表している．

例えば，図 15 において，入力系列の季節は「夏」であるのにも関わらず，出力系列の季節は「春」となってしまった俳句は 171 句ある．また，入力系列，出力系列両方において季節が「夏」だった俳句，つまり入力系列の季節を出力系列でも反映できている俳句は 1 句である．ここで，図 15，図 16 では季節素性や季節制約に関する情報を持たせずに生成した結果であるため，入力時の単語の季節が出力結果に必ずしも反映されているとは限らない．

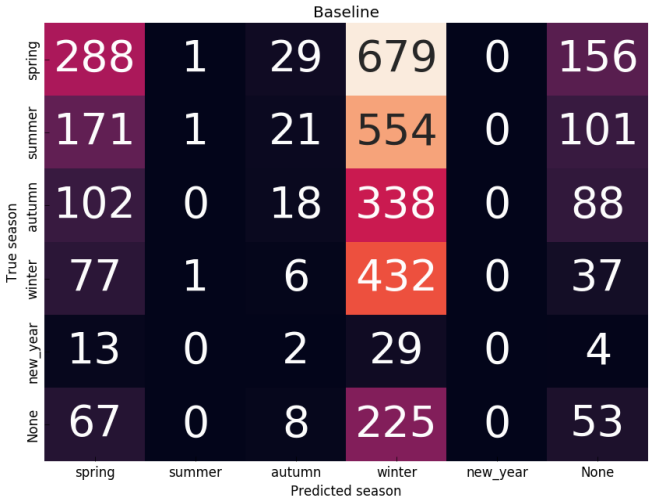

図 15　Baseline

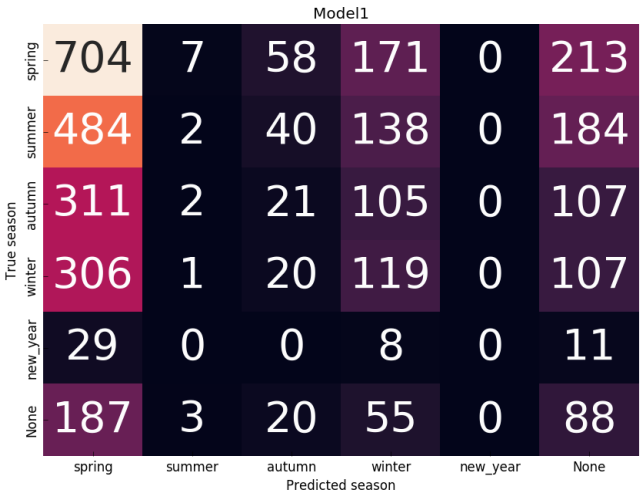

図 16　Model 1

Vol.2018-NL-235 No.1
Vol.2018-SLP-121 No.1
2018/5/13

図 17　Model2

図 18　Model3

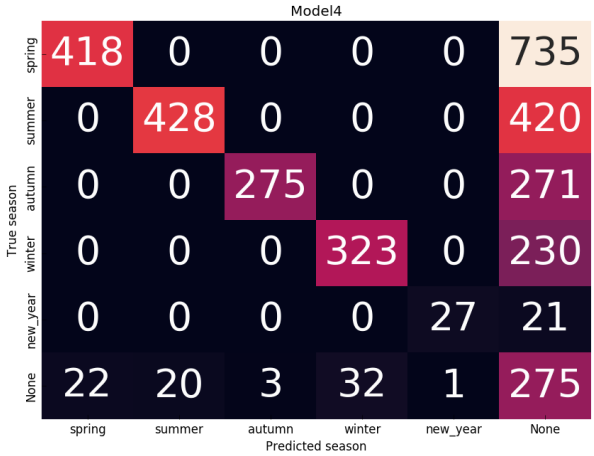

図 19　Model4

図 15 や図 16 と図 17～図 19 を比較すると，季節の素性や制約を加えると．入力系列の季節の季語を出力することが出来ている俳句が多いことが分かる．一方で，季節の素性を加えているのに季語がないという出力も増えている．これは，拍数の素性を加えた場合と同じように，素性や制限を加えた事で選べる単語の幅が狭まってしまうからだと考えられる．図 18 と図 19 を比較すると，図 19 は入力系列が無季節・無季語の場合を除いては，入力系列と異なる季節を出力される事はなくなった．一方でこの強い制約を入れた事で，より単語の選択肢が狭まり，入力系列に季節や季語が含まれているのにも関わらず，季語が含まれないような出力が増加した．

6. おわりに

本研究では，有季定型俳句の特徴に基づいて，拍数の素性や季語の素性、季節の制約を注意機構付きの系列変換モデルに加えて比較検討した．拍数の素性を導入した方が正しい韻律で生成されることが確認された．また，季節の素性を加えた方が，入力系列の季節と同じ季節の俳句が生成されやすいということが分かった．さらに，季語制約を加えると，入力系列の季節以外の季節の季語は生成されないということが確認された．

しかし課題点も残っている．まず，季節の素性や制約を加えて出力したい季節を出力させやすくしたり，出力したい季節以外を出力させないようにすることは出来たが，出力したい季節を必ず出力させることはできていない．必ず出力させるために，生成中に出力したい季節の季語が出るまで単語を選ばせるという方法が考えられるが，その場合は生成された俳句の上五，中七，下五のいずれかに季語が固まってしまう可能性がある．これでは，柔軟な表現の俳句が生成できているとは言いがたいと考え，今回の実験ではそのような手法は実装しなかった．

次に，拍数素性以外に季節素性や制約を導入した際に選べる単語の幅が狭まり，拍数素性のみを導入した場合に比べて正しい拍数で生成された俳句の数が少なくなってしまっているという問題点が挙げられる．

さらに，今回は切れや切れ字に関して何も処理を施していない．それは切れや切れ字は出現する場所は決まっていないが俳句に必ず 1 つ以上含まれているので，深層学習で切れや切れ字が出現しそうな場所に生成されることを期待したからである．しかし，切れや切れ字が含まれていない俳句も見受けられたので，何らかの処理を施す必要がある．

最後に，今回は生成した俳句の評価は，季語の有無と拍数のみでの評価を行った．俳句や漢詩の自動生成に関する先行研究では，翻訳の分野の指標を用いて評価している研究が幾つか見受けられた．しかし本研究では，良い俳句は 1 つに決まらず，正解がある翻訳とは異なると考え，翻訳の指標で俳句を評価するということはしなかった．本来，良い俳句とは季語の有無や拍数だけで判定されるものではなく，主観評価によっても判定される．良い俳句であるか否かは，俳句に慣れ親しんでいない人間には判定しづらい．現に，今回のデータセットは有名な俳人の俳句によって構

成されているが，中には素人の自分でも詠めそうな俳句もある．俳句に詳しい方に依頼して，機械で生成された俳句と俳人の俳句を見比べてもらい，どちらがより良い俳句であるか判定してもらったりするなどの主観評価が必要である．

謝辞 本論文を完成するにあたり，奈良先端科学技術大学院大学の自然言語処理学研究室の濱口さん，大内さんをはじめ，同研究室の多くの方にお世話になりました。深く感謝いたします．

参考文献

[1] Naoko Tosa, Hideo Obara, Michihiko Minoh. ``Hitch haiku: An interactive supporting system for composing haiku poem." .In *Proceedings of ICEC*, vol.5309, pp.209-216, 2008.
[2] Xiaofeng Wu, Naoko Tosa, Ryohei Nakatsu. ``New Hitch Haiku: An Interactive Renku Poem Composition Supporting Tool Applied for Sightseeing Navigation System." In *ICEC*, pp. 191-196, 2009.
[3] Xianchao Wu, Momo Klyen, Kazushige Ito, Zhan Chen. ``Haiku Generation Using Deep Neural Networks", 2017. 言語処理学会
[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. ``Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." In *EMNLP* 2014.
[5] Lantao Yu,weinan Zhang, Jun Wang, Yong Yu. ``SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient." In *AAAI* 2017.
[6] Yoon Kim, Yacine Jernite, David SOntag, and Alexander M. Rush. `` Character-aware neural language models." In *AAAI*, 2016.
[7] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. ``Handwritten Digit Recognition with a Back-Propagation Network."In *Proceedings of NIPS,* 1989.
[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza,B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. ``enerative adversarial nets." In *Proceedings of NIPS,* pp. 2672–2680, 2014.
[9] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. ``Sequence to Sequence Learning with Neural Networks." In *Proceedings of NIPS*, pp. 3104–3112, 2014.
[10] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. ``Neural Machine Translation by Jointly Learning to Align and Translate." In *Proceedings of ICLR*, 2015.
[11] Minh-Thang Luong, Hieu Pham, Christopher D. Manning. ``Effective Approaches to Attention-based Neural Machine Translation." In *Conference on Empirical Methods in Natural Language Processing*, 2015.
[12] S. Hochreiter and J. Schmidhuber. ``Long short-term memory." *Neural Computation*, 1997.
[13] Marjan Ghazvininejad, Xing Shi, Yejin Choi, Kevin Knight. ``Generating Topical Poetry." In *Proceedings of EMNLP*, 2016.
[14] J.Kao, D. Jurafsky. ``A Computational Analysis of Style, Affect, and Imagery in Contemporary Poetry." In *NAACL Workshop on Computational Linguistics for Literature*, 2012.
[15] Jing He, Ming Zhou, and Long Jiang. ``Generating Chinese Classical Poems with Statistical Machine Translation Models." In *Proceedings of the AAAI*, pp. 1650-1656, 2012.
[16] Zhe Wang, Wei He, Hua Wu, Haiyang Wu, Wei Li, Haifeng Wang, Enhong Chen. ``Chinese Poetry Generation with Planning based Neural Network."In *COLING*, 2016.
[17] Qixin Wang, Tianyi Luo, Dong Wang, and Chao Xing. ``Chinese song iambics generation with neural attention-based model." *arXiv:1604.06274*. 2016.
[18] Xingxing Zhang, Mirella Lapata. ``Chinese Poetry Generation with Recurrent Neural Networks." In *Proceedings of EMNLP*, 2014.
[19] 一茶の俳句データベース http://ohh.sisos.co.jp/cgi-bin/openhh/jsearch.cgi?group=hirarajp
[20] 芭蕉発句全集 http://www2.yamanashi-ken.ac.jp/~itoyo/basho/haikusyu/Default.htm

付録

生成結果例

実際に Model4 で生成された俳句を以下に示す．

[入力系列] 春 山 風
[Baseline] 花の ｜ 人の中に ｜ 風の蝶
[Model1] 一本の ｜ 水の底たる ｜ 夜寒かな
[Model2] 春雨や ｜ 大中の中 ｜ 山の秋
[Model3] 陽炎の ｜ 花の女の ｜ 春の雨
[Model4] 鶯の ｜ 口を被って ｜ 善光寺