NAIST-IS-MT1651023

修士論文

深層学習による俳句の自動生成

太田 瑶子

平成 30 年 3 月 14 日

奈良先端科学技術大学院大学
情報科学研究科 情報科学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士（工学）授与の要件として提出した修士論文である.

太田 瑶子

審査委員：

松本 裕治 教授	（主指導教員）
中村 哲 教授	（副指導教官）
新保 仁 准教授	（副指導教官）
進藤 裕之 助教	（副指導教官）
能地 宏 助教	（副指導教官）

深層学習による俳句の自動生成*

太田 瑶子

内容梗概

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．その場の上手く言い表しきれないような微妙な心情や情景は，長々しく文を綴って記録することも出来る．しかし，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．詩の作り方や鑑賞方法は義務教育課程の授業で習う．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，　詠みたい内容の語や季語を入力することにより俳句の自動生成を行った．先行研究では，ルールベースで俳句を生成しているものや，データ数を増やすために季語の有無は考慮していないデータを用いて深層学習で生成を試みた研究がある．本研究ではより柔軟な表現ができるように，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．

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

*奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 修士論文, NAIST-IS-MT1651023, 平成 30 年 3 月 14 日.

Haiku Generation with Deep Learning*

Yoko Ota

Abstract

Poetry expresses things of nature and feelings in human life as rhythmic linguistic form. The message is conveyed not only by the superficial meaning of words, but also by aesthetic and envocative 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 exists 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 use deep learning to make more flexible representation. We also added haiku's features to deep learning model.

Keywords:

haiku, automatically generation, machine-learning, neural networks

*Master's Thesis, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651023, March 14, 2018.

目 次

図 目 次 **v**

第 1 章　はじめに **1**
1.1 背景 1
1.2 本論文の構成 2

第 2 章　計算機による俳句の自動生成 **3**
2.1 俳句について 3
2.2 先行研究 6
2.2.1 ルールに基づいた俳句創作支援システム 6
2.2.2 深層学習を使った俳句自動生成 6
2.2.3 本研究と先行研究の違い 8

第 3 章　提案手法 **9**
3.1 本研究の基本となる技術の紹介 9
3.1.1 系列変換モデル (sequence-to-sequence model; seq2seq モデル) 9
3.1.2 注意機構 (attention mecanism) 10
3.2 モデルの基本構造 10
3.2.1 Encoder 部分 10
3.2.2 Decoder 部分 11
3.3 拍数素性の導入 13
3.4 季節素性の導入 14
3.5 季節制限 14

第 4 章　実験 **16**
4.1 データセットについて 16
4.2 結果・考察 17
4.2.1 拍数素性の有無による結果 17
4.2.2 季節素性の導入と季語制限による結果 23

第 5 章　おわりに **29**

謝辞 **30**

参考文献 **31**

付録 **33**

図目次

1.1 定型詩の例 (与謝野晶子『恋衣』より) 1
1.2 自由詩・散文詩の例 (中原中也『山羊の歌』より) 1

2.1 俳句の特徴 (松尾芭蕉『春の日』より) 3
2.2 自由律俳句の俳句例 (尾崎放哉『大空』より) 4
2.3 字足らずの俳句例 (尾崎放哉『大空』より) 4
2.4 字余りの俳句例 (高浜虚子『五百句』より) 4
2.5 季重ねの俳句例 (山口素堂 作) 4
2.6 季違いの俳句例 (小林一茶 作) 5
2.7 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) 5
2.8 句またがりの俳句例 (松尾芭蕉『野ざらし紀行』より) 5

3.1 seq2seq モデル 9
3.2 注意機構の構造 [11] より引用 12
3.3 拍数素性の導入 13
3.4 季節素性の導入 14

4.1 拍数の比較 (全体) 17
4.2 拍数の比較 (上五) 18
4.3 拍数の比較 (中七) 20
4.4 拍数の比較 (下五) 22
4.5 Baseline 24
4.6 Model1 25
4.7 Model2 26
4.8 Model3 27
4.9 Model4 28

第1章　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．

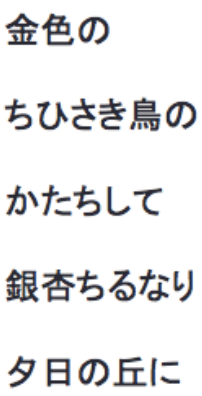

図 1.1: 定型詩の例 (与謝野晶子『恋衣』より)

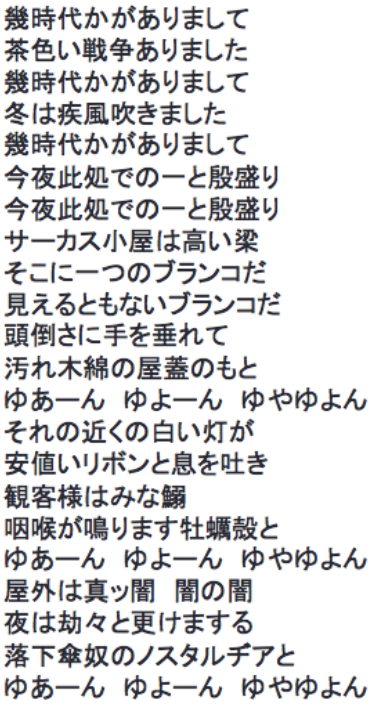

図 1.2: 自由詩・散文詩の例 (中原中也『山羊の歌』より)

図 1.1 の詩は，金色に輝いている小鳥を思わせるような銀杏の葉が，夕日の綺麗な丘で散っている情景を表現した詩である．これは詩の中でも短歌と呼ばれるものであるが，図のように 5 つのフレーズに分けられ，韻律のあるような定型詩である．一方，図 1.2 の詩では，特に決まった規定はない．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難しく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 本論文の構成

本論文の構成について述べる．本論文では，まず第 2 章で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．第 3 章では，提案手法とその基本となる技術について述べる．第 4 章では，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．

第2章　計算機による俳句の自動生成

本章では，俳句についての説明及び，計算機による俳句の自動生成についての先行研究について説明する．

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．

1. 上五，中七，下五はそれぞれ 5 拍，7 拍，5 拍
2. 1 俳句につき，季節を表す 1 語 (季語)
3. 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

松尾芭蕉が詠んだ俳句を例として俳句の特徴を図 2.1 に示す．

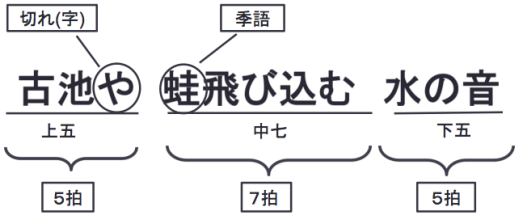

図 2.1: 俳句の特徴 (松尾芭蕉『春の日』より)

図 2.1 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 (図 2.2).

図 2.2: 自由律俳句の俳句例 (尾崎放哉『大空』より)

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 2.3，図 2.4 参照)

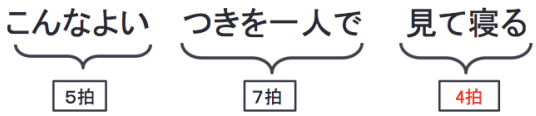

図 2.3: 字足らずの俳句例 (尾崎放哉『大空』より)

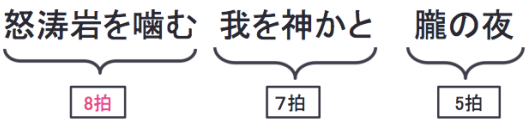

図 2.4: 字余りの俳句例 (高浜虚子『五百句』より)

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 2.5，図 2.6 参照)

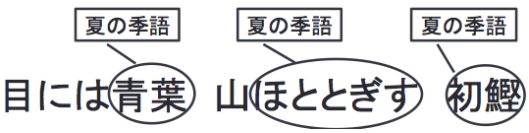

図 2.5: 季重ねの俳句例 (山口素堂 作)

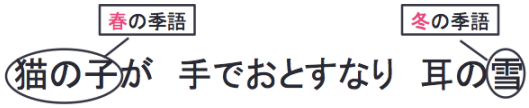

図 2.6: 季違いの俳句例 (小林一茶 作)

図 2.5 の俳句には夏の季語が 3 つ入っており，季重ねである．図 2.6 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．季違いは隣同士の季節だけでなく，図 2.7 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 2.7: 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 2.6 や図 2.7 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 2.6 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 2.7 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．

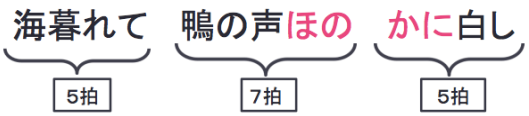

図 2.8: 句またがりの俳句例 (松尾芭蕉『野ざらし紀行』より)

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 2.8 では,「ほのかに」という語が中七と下五をまたいでいる．

本研究では，一番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] の研究で，コンピューターによる俳句創作支援システム (Hitch Haiku) である． 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い応じて，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．

1. vanilla Recurrent Neural Network (RNN) モデル
2. Long Short-Term Memory (LSTM) を使ったモデル
3. Kim ら [6] の Recurrent convolutional neural networks (RCNN) モデル
4. 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，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 層にベクトルに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．データセットについては，次の表 2.1 に示す．

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．

表 2.1: 2 種類のデータセット

	データ	俳句の数	語彙サイズ
チャットボット	train	90,000	2,934
	validation	5,000	1,353
	test	5，000	1,347
Web サイト	train	30,792	4,110
	validation	3,000	2,416
	test	3,000	2,435

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．

本研究では，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．

第 3 章　提案手法

この章では，本研究の基本となる技術を紹介し，その技術を使ったモデルの構造について説明する．最後に，提案手法を述べる．なお，本研究では，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　古池 蛙　飛び 込む 水 音

出力系列　古池 や | 蛙 飛び 込む | 水 の 音

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

3.1　本研究の基本となる技術の紹介

研究に関するベースとなる技術を以下に説明する．

3.1.1　系列変換モデル (sequence-to-sequence model; seq2seq モデル)

系列 (sequence) を入力すると，別の系列に変換する確率をモデル化したものを系列変換モデル (sequence-to-sequence model; seq2seq モデル [9]) という．文字列や単語列が系列に相当する．図 3.1 では，“ ABC ”という系列が入力文であり，それを 1 つずつエンコーダー (Encoder) で読み込む．そして入力文が変換された系列として“ ABC ”に対応するような“ WXYZ ”という系列がデコーダー (Decoder) から生成されるように学習する．ここで，“ <EOS> ”は“ End Of Sentence ”を表す記号で，文末であることを意味している．

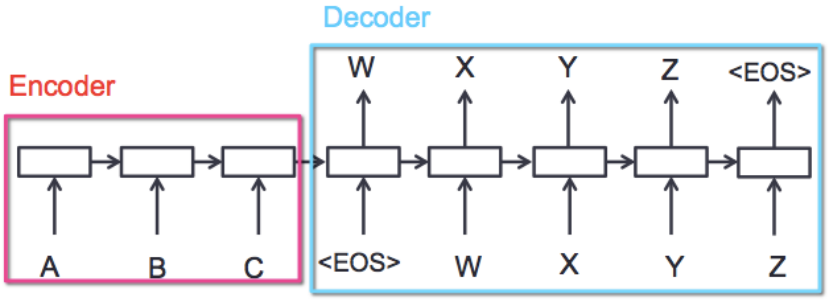

図 3.1: seq2seq モデル

図 3.1 の各ボックスは，本研究では LSTM[12] のセルである．

3.1.2 注意機構 (attention mecanism)

系列変換モデル単体では，LSTM を使っても最初に入力した情報を最後まで伝える事が難しい．そこで，系列変換モデルに注意機構 (attention mechanism[10]) を入れる．注意機構の中には，ソフト注意機構とハード注意機構というものがある．入力系列の各単語ベクトルをエンコードした際の隠れ層のベクトルをそれぞれ保存しておいて，デコード時にどの隠れ層のベクトルをどのくらい重み付けすべきか学習させるものがソフト注意機構である．一方，ハード注意機構は，見るべき隠れ層ベクトルを 1 つに決めてしまう方法である．本研究ではソフト注意機構を用いた．

この章では提案手法について説明する．

3.2 モデルの基本構造

注意機構の付いた系列変換モデルを使用する．以下、Encoder 部分と Decoder 部分に分けて説明する．

3.2.1 Encoder 部分

Encoder は埋め込み (embedding) 層と再帰層に分けられる．まず，埋め込み層では，Encoder で受け取った入力系列の第 i 番目の単語の one-hot ベクトル x_i，それに対応する埋め込みベクトルを $\bar{x_i}$ とすると，x_i から $\bar{x_i}$ への変換は埋め込み行列 $\boldsymbol{E}$ を使って以下の式で表せる．

$$\boldsymbol{\bar{x}_i} = \boldsymbol{E}\boldsymbol{x_i} \tag{3.2.1}$$

再帰層では LSTM を使って，入力したい系列を固定長の隠れ状態ベクトル h に変換する．LSTM の内部の構造には，入力 (input) ゲート $\boldsymbol{i}$，忘却 (forget) ゲート $\boldsymbol{f}$，出力 (output) ゲート $\boldsymbol{o}$ の 3 つのゲートがある．LSTM は時刻 t における入力 $\boldsymbol{x_t}$ を受け取ると，時刻 $t-1$ における隠れ層状態ベクトル $\boldsymbol{h_{t-1}}$ と $\boldsymbol{x_t}$ を使って内部で計算し，隠れ状態ベクトル $\boldsymbol{h_t}$ を求める．

$$\boldsymbol{h_t} = \boldsymbol{o_t} \otimes \tanh(\boldsymbol{c_t}) \tag{3.2.2}$$

ここで $\boldsymbol{c}$ とは，隠れ状態ベクトル $\boldsymbol{h}$ の他に長期の情報を保持するためのセルと呼ばれる隠れ状態ベクトルである．出力ゲートは次のように計算される．

$$\boldsymbol{o_t} = \sigma(\boldsymbol{W_o x_t} + \boldsymbol{U_o h_{t-1}} + \boldsymbol{b_o}) \tag{3.2.3}$$

入力ゲート，忘却ゲート，セルは以下のように計算される．

$$\boldsymbol{i_t} = \sigma(\boldsymbol{W_i x_t} + \boldsymbol{U_i h_{t-1}} + \boldsymbol{b_i}) \tag{3.2.4}$$

$$\boldsymbol{f_t} = \sigma(\boldsymbol{W_f x_t} + \boldsymbol{U_f h_{t-1}} + \boldsymbol{b_f}) \tag{3.2.5}$$

$$\boldsymbol{c_t} = \boldsymbol{c_{t-1}} \otimes \boldsymbol{f_t} + \boldsymbol{i_t} \otimes \tanh(\boldsymbol{W_c x_t} + \boldsymbol{U_c h_{t-1}} + \boldsymbol{b_c}) \tag{3.2.6}$$

3.2.2 Decoder 部分

Decoder は，埋め込み層，再帰層，出力層に分けられる．まず，埋め込み層では出力層で出力された y_{j-1} と埋め込みベクトル $\bar{y_j}$ を用いて次のように表すことが出来る．

$$\bar{\boldsymbol{y}_j} = \boldsymbol{E y_{j-1}} \tag{3.2.7}$$

再帰層では，本研究においては LSTM を使用する．

ここで，本研究で使っている注意機構について説明する．注意機構では，Encoder の各時刻の隠れ状態ベクトル $\bar{\boldsymbol{h}_i}$ と時刻 t における Decoder の隠れ状態ベクトル $\boldsymbol{h_t}$ を使って，どの入力系列に注目するべきかというスコア α_t を決定する．

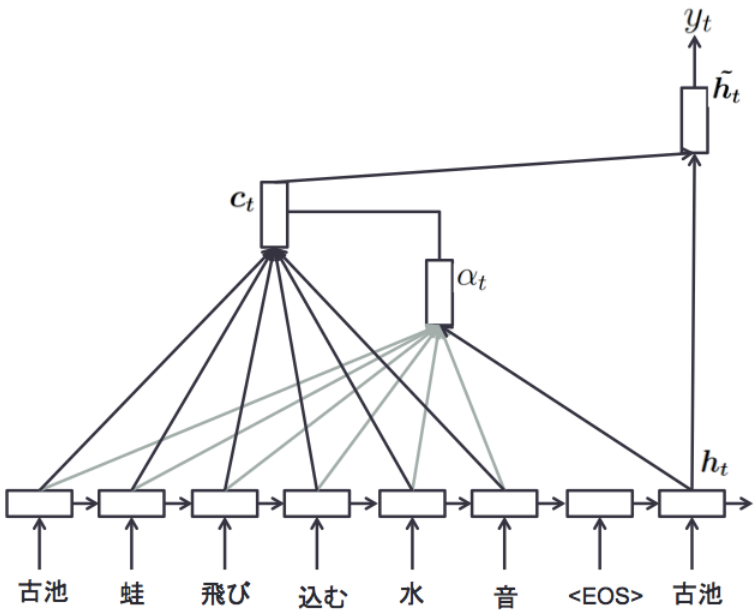

図 3.2: 注意機構の構造 [11] より引用

$$\alpha_t(i) = \frac{exp(\boldsymbol{h_t}, \bar{\boldsymbol{h_i}})}{\sum_{j=1} exp(\boldsymbol{h_t}, \bar{\boldsymbol{h_i}})} \tag{3.2.8}$$

この $\alpha_t(i)$ を元に，Encoder の隠れ状態ベクトルを加重平均したベクトル $\boldsymbol{c_t}$ を以下のように求める．

$$\boldsymbol{c_t} = \sum_{i=1} \alpha_t(i) \bar{\boldsymbol{h_i}} \tag{3.2.9}$$

$\boldsymbol{c_t}$ と $\boldsymbol{h_t}$ を使用して，Decoder の隠れ層の出力 $\tilde{\boldsymbol{h_t}}$ を計算する．

$$\tilde{\boldsymbol{h_t}} = \tanh(\boldsymbol{W_c}[\boldsymbol{c_t}; \boldsymbol{h_t}]) \tag{3.2.10}$$

その後，出力層に移るのだが，ここでは学習の場合とテスト (test) の場合でタイプが分かれる．学習時は訓練データと現在のモデルが適合しているか判断するために，$\tilde{\boldsymbol{h_t}}$ を用いて，y_t の生成確率を計算する．

$$P(y_t|Y_{<t}) = \mathsf{softmax}(\boldsymbol{W_o}\tilde{\boldsymbol{h_t}} + \boldsymbol{b_o}) \cdot \boldsymbol{y_t} \tag{3.2.11}$$

テスト時は，Decoder から生成される単語 $\hat{y}_t$ を計算する．

$$\hat{\boldsymbol{y}}_{\boldsymbol{t}} = \mathsf{softmax}(\boldsymbol{W_o}\tilde{\boldsymbol{h_t}} + \boldsymbol{b_o}) \tag{3.2.12}$$

3.3　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，Decoder で入力された単語ベクトルに，拍数の素性の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 進数で表している．

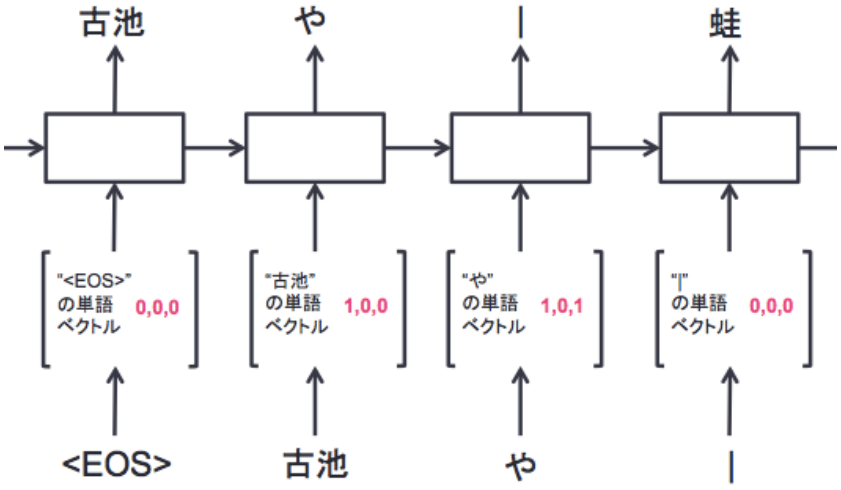

図 3.3: 拍数素性の導入

図 3.3 では，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 3.3 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．

3.4　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．

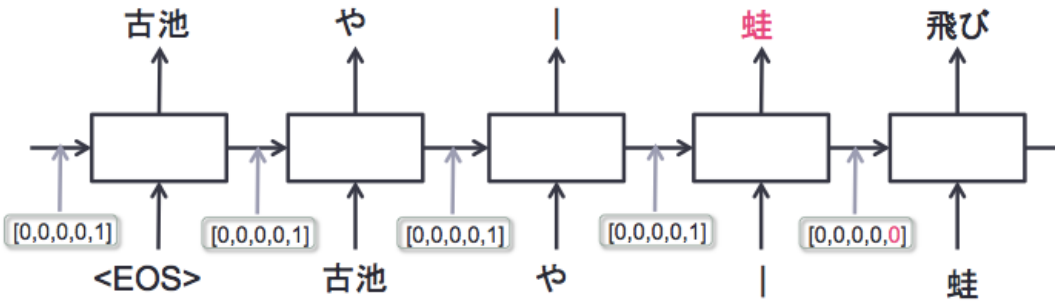

図 3.4: 季節素性の導入

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 3.4 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 3.4 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．

3.5　季節制限

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限を加えた．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限は加えていない．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．

第4章　実験

4.1　データセットについて

本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．以下の表4.1に示す．

表 4.1: 実験で用いたのデータセット

	データ数	各季節の俳句数						語彙サイズ
		春	夏	秋	冬	新年	無	
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，以下に示すような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．

季語のデータセットについて以下の表4.2に示す．

表 4.2: 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 attention seq2seq

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，季語制限を加えたモデル

4.2 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，季節素性と季語制限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．

4.2.1 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 4.1 に示す．

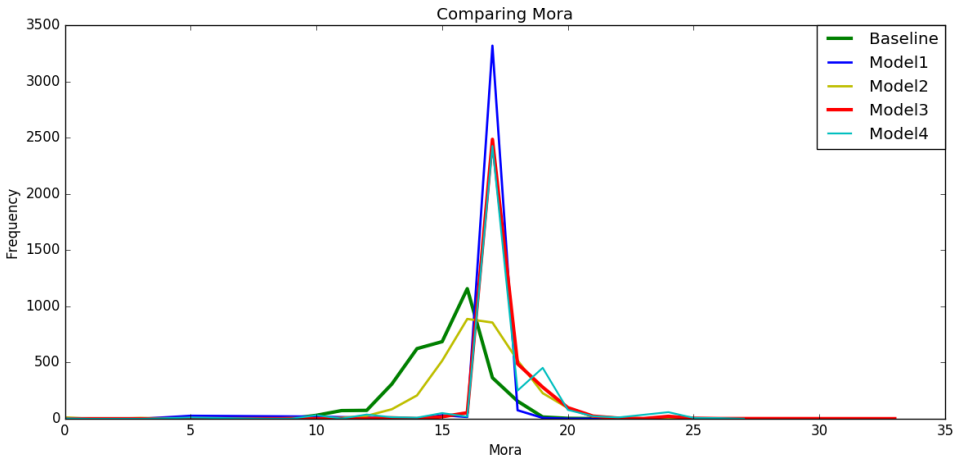

図 4.1: 拍数の比較 (全体)

表 4.3: 拍数の平均と分散 (俳句全体)

	baseline	model1	model2	model3	model4
平均	15.5	16.9	16.4	17.4	17.3
分散	2.55	1.46	4.42	1.35	3.71

図 4.1 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 4.1 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．

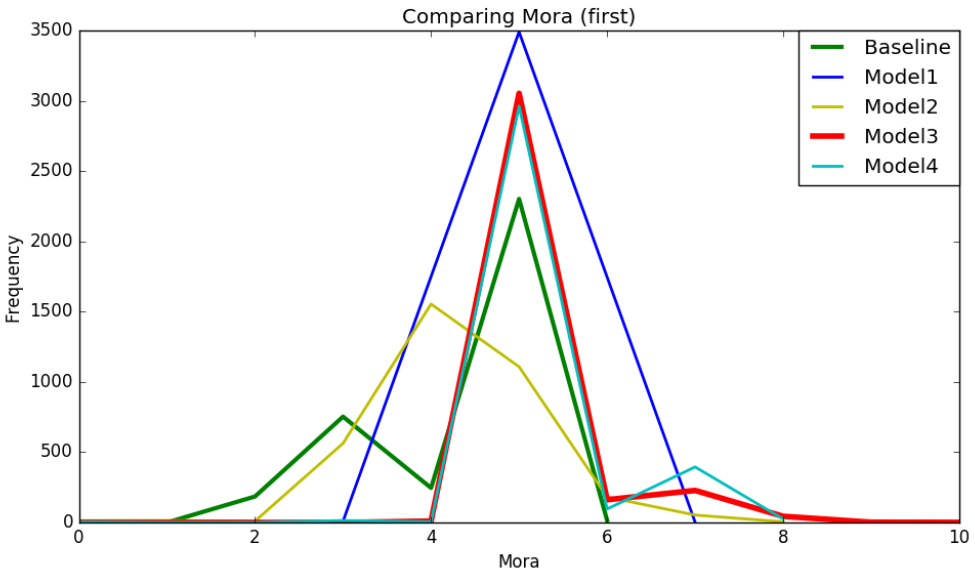

図 4.2: 拍数の比較 (上五)

表 4.4: 拍数の度数分布表 (上五)

拍数	頻度				
	baseline	model1	model2	model3	model4
0	0	0	13	1	4
1	0	0	14	0	0
2	185	0	10	0	0
3	753	9	563	0	14
4	246	0	1,554	10	5
5	2,302	3,490	1,108	3,053	2,962
6	15	0	181	162	96
7	0	2	53	228	395
8	0	0	5	44	25
9	0	0	0	2	0
10	0	0	0	1	0

表 4.5: 拍数の平均と分散 (上五)

	baseline	model1	model2	model3	model4
平均	4.35	5.00	4.28	5.21	5.26
分散	0.982	0.0126	0.872	0.401	0.522

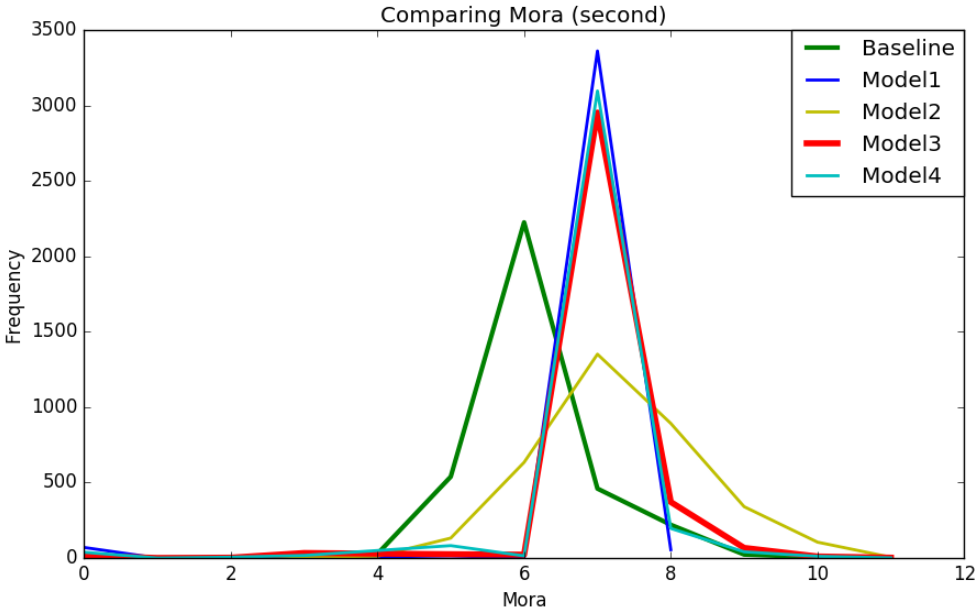
Comparing Mora (second)
Baseline
Model1
Model2
Model3
Model4
Frequency
Mora
0
500
1000
1500
2000
2500
3000
3500
0
2
4
6
8
10
12

図 4.3: 拍数の比較 (中七)

表 4.6: 拍数の度数分布表 (中七)

拍数	頻度				
	baseline	model1	model2	model3	model4
0	0	70	24	16	38
1	0	0	0	0	0
2	0	0	0	2	5
3	0	0	6	33	17
4	27	0	12	0	0
5	539	14	132	25	82
6	2,225	1	634	22	14
7	460	3,361	1,351	2,953	3,096
8	220	55	892	371	196
9	23	0	340	67	41
10	5	0	105	10	11
11	2	0	5	2	1

表 4.7: 拍数の平均と分散 (中七)

	baseline	model1	model2	model3	model4
平均	6.12	6.87	7.22	7.06	6.94
分散	0.650	0.994	1.61	0.637	0.874

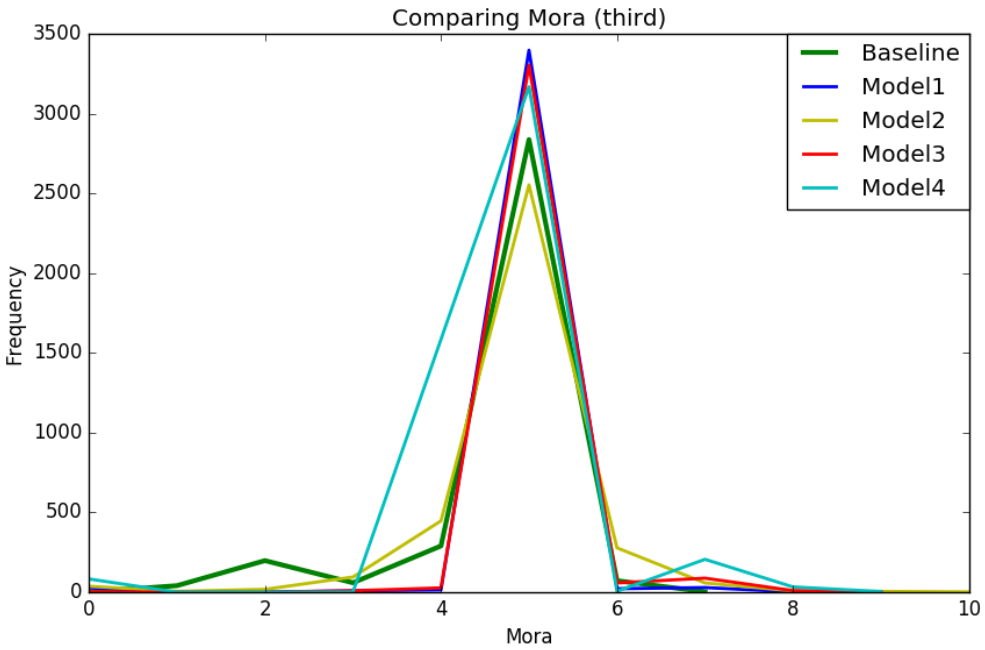
Comparing Mora (third)
Baseline
Model1
Model2
Model3
Model4
Frequency
Mora
0
500
1000
1500
2000
2500
3000
3500
0
2
4
6
8
10

図 4.4: 拍数の比較 (下五)

表 4.8: 拍数の度数分布表 (下五)

拍数	頻度				
	baseline	model1	model2	model3	model4
0	1	27	37	3	83
1	41	0	5	0	0
2	198	1	18	0	0
3	59	9	95	9	1
4	290	14	447	27	0
5	2,837	3,397	2,553	3,303	3,168
6	72	22	278	58	6
7	3	29	58	88	206
8	0	2	8	10	33
9	0	0	0	3	4
10	0	0	2	0	0

表 4.9: 拍数の平均と分散 (下五)

	baseline	model1	model2	model3	model4
平均	4.69	4.98	4.87	5.06	5.03
分散	0.780	0.254	0.732	0.192	0.933

図 4.2 から図 4.4 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 4.3 と，5 拍が正しい拍数である図 4.2，図 4.4 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．

4.2.2 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限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使って以下に示した．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 4.5 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．

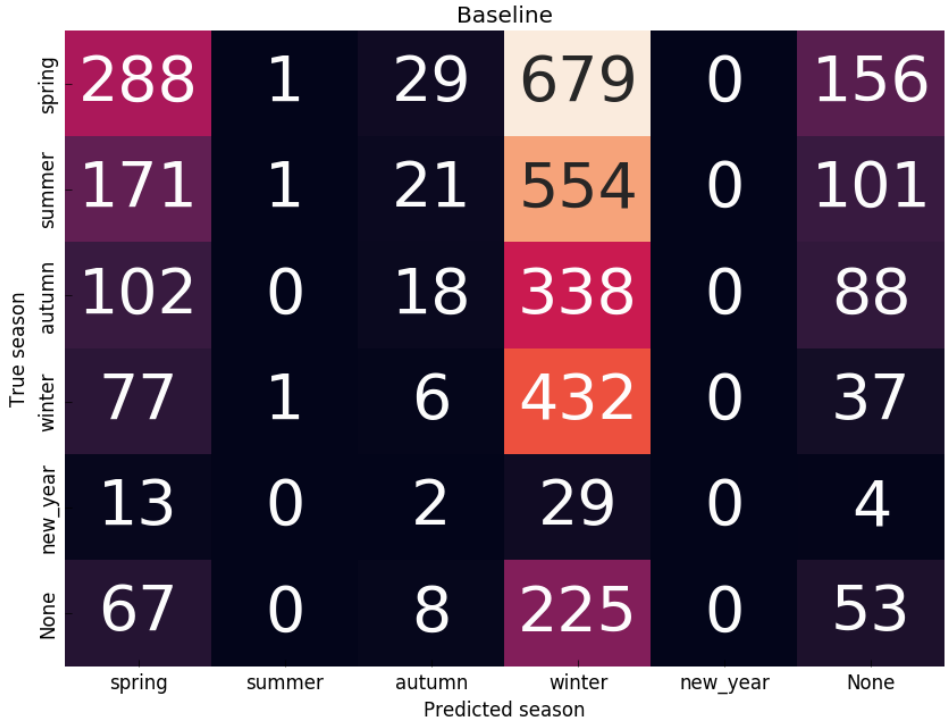

図 4.5: Baseline

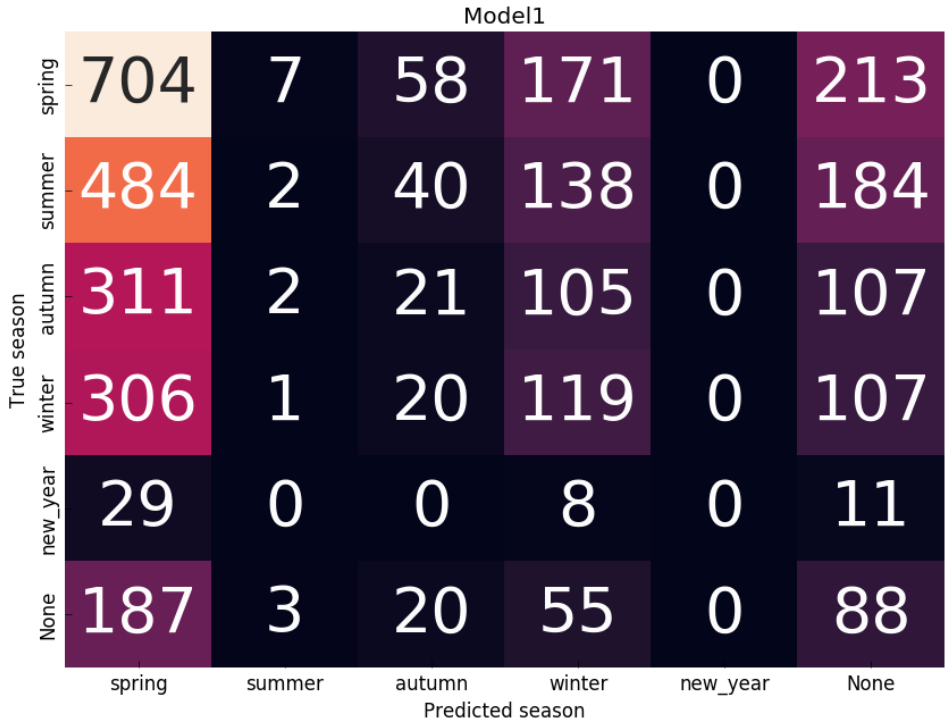
Model1
True season
spring
summer
autumn
winter
new_year
None
704 7 58 171 0 213
484 2 40 138 0 184
311 2 21 105 0 107
306 1 20 119 0 107
29 0 0 8 0 11
187 3 20 55 0 88
spring
summer
autumn
winter
new_year
None
Predicted season

図 4.6: Model1

Model2
True season
spring
summer
autumn
winter
new_year
None
623 39 38 23 1 429
42 414 35 18 1 338
32 21 268 8 0 217
30 35 31 237 1 219
2 4 6 1 24 11
16 7 6 2 0 322
spring
summer
autumn
winter
new_year
None
Predicted season

図 4.7: Model2

Model3
True season
spring
summer
autumn
winter
new_year
None
637 36 24 31 6 419
54 453 22 15 11 293
38 28 246 14 22 198
74 44 24 197 4 210
3 19 2 0 19 5
25 26 10 11 1 280
spring
summer
autumn
winter
new_year
None
Predicted season

図 4.8: Model3

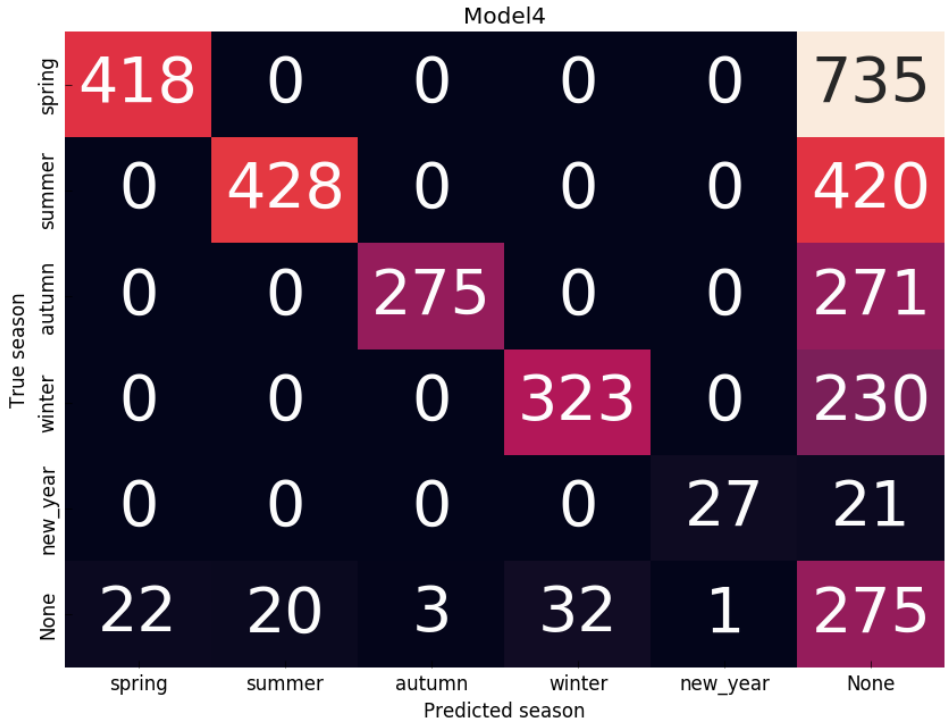

図 4.9: Model4

図 4.5 や図 4.6 と図 4.7〜図 4.9 を比較すると，季節の素性や制限を加えると．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 4.8 と図 4.9 を比較すると，図 4.9 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．一方でこの強い制限を入れた事で，より単語の選択肢が狭まり，入力系列に季節や季語を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えた．

最後に，実際にそれぞれのモデルから出力された俳句については付録に一部掲載した．

第5章　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、季節の制限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，季節制限を加えると，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，季節の素性や制限を加えて出力したい季節を出力させやすくさせたり，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ものが考えられるが，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，今回の実験では行わなかった．次に，拍数素性以外に季節素性や制限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

はじめに，本研究を行うにあたり，主指導教員である自然言語処理学講座の松本裕治教授に深く感謝を申し上げます．沢山ご迷惑おかけしたにも関わらず，最後まで見放さずにご指導くださいまして本当にありがとうございました．周りと比べ過ぎて，自分は何も出来ないと思い詰めていました．ゼミナールでアドバイスをくださった知能コミュニケーション研究室の中村哲教授，新保仁准教授，相談に乗ってくださった進藤裕之助教や能地宏助教，博士後期課程の濱口さんや大内さん，いつも気に掛けてくださった秘書の北川さん，他にも研究室の方々に心から感謝致します．また寮の仲間や遠方の友人，ずっと一緒に勉強してくれていた知能コミュニケーションの友人など支えてくださいまして，ありがとうございました。最後に，小言が煩くて辛かった時もありましたが，いつも応援してくれていた家族に感謝いたします．

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.

付録

入力系列を俳句の内容語として俳句を生成した．正解の俳句は「古池や | 蛙飛び込む | 水の音」である．

入力系列 古池 蛙 飛び込む 水 音

季節 春

Baseline みな弦の | 蛙蛙も | 水の音

Model1 春雨や | 水の中より | 水の音

Model2 夜は後 | 古池蛙それ | 水の音

Model3 炎天の | さして蛙の | 萩の音

Model4 たき万の | 古池の夕 | 水の音

正解の俳句 古池や | 蛙飛び込む | 水の音

以下では，季語と語をいくつか並べて入力し生成した．よって，以下の入力系列には正解の俳句はない．

入力系列 アイスクリーム 買う 小遣い

季節 夏

Baseline 紐ばや | 夜の馬の | 買う移り

Model1 冬の葉の | 中にもなれて | 冬払い

Model2 やしもや | これやれ熱く | 買うろり

Model3 夏のごの | 錠の家族の | 置いて佇つ

Model4 結び目の | 造るとて買う | て恥しき

入力系列 銀杏 道 黄色

季節 秋

Baseline 祈りの | つつ銀杏づして | 道すべ

Model1 ふるさとや | 人の字をいる | 見てもなし

Model2 夜よりも | シカと銀杏 | 道を

Model3 稲のから | あいその火事の | 道の夕

Model4 秋風の | それだけの柄の | 道をけり

. . . . . .

入力系列 朝 雪 窓 外

季節 冬

Baseline 村までは | 夜の朝雪の | 窓に外

Model1 人の | 一つの中の | 水の蝶

Model2 夜からしも | 朝朝雪の | 窓と外

Model3 たまにもも | 朝の雪雪 | 窓の原

Model4 結び目の | 朝の朝窓 | 窓の鱈

. . . . . .

入力系列 初句会 着物 歩く

季節 新年

Baseline てつかんな | よりゴミを着物 | 着物歩く

Model1 春の世に | 水の字の日に | 一つする

Model2 世からへ | へばめいし | 着物歩く

Model3 火の見れに | 似てそれぞれの | 落花かな

Model4 何を語る | 日々の着物に | 銭歩く