Vol.2018-ICS-192 No.8
2018/7/6

深層学習による生成句判別

米田 航紀[1,a)]　横山 想一郎[1,b)]　山下 倫央[1,c)]　川村 秀憲[1,d)]

概要：
深層学習を使用した芸術の作成が近年注目を集めている.
また, 日本で古くから親しまれている芸術として俳句がある.
そこで, 俳句を作成する方法として一般的な「モチーフから俳句を作る」ということを深層学習を使用して行うことで, 芸術作成としての深層学習の有用性を示す. まず, 我々は大量の過去の俳句に基づいて LSTM を訓練し, LSTM に文字列を生成させる.
2 つめに, 生成された文字列から俳句としての条件を満たすものを抽出する.
3 つめに，得られた俳句の質を評価するために深層学習を使用し, 人間が作成した俳句と本研究で作成した俳句の判別器を作成し評価する.
評価値が高ければ, 生成された俳句が質の高い良い俳句とみなす.
この過程で, LSTM が俳句としてのルールを学習できているか，評価用の判別器は正しく評価できるかを確認するための実験を行った.

Yoneda Koki[1,a)]　Yokoyama Soichiro[1,b)]　Yamashita Tomohisa[1,c)]　Kawamura Hidenori[1,d)]

1. はじめに

1.1 研究背景

近年, 深層学習の研究の発展により文章の翻訳や, 画像のキャプション生成など様々な場面で使用されている.

画像のキャプション生成の研究としては,Google の研究チームが発表した研究がある. これは Convolutional Neural Network (CNN) を使用している研究である. この研究では画像を Deep CNN によって特徴ベクトルに変換し, それを翻訳用の Recurrent Neural Network (RNN)[3] を使用して文章に変換する. この時に使用している RNN は Google 翻訳で使用されている [4]. この研究ではキャプションの性能について定性的及び定量的に堅牢な文章作成が可能だとしている.

また, 中でも注目を集めている研究として絵などの芸術の生成がある. この分野では DCGAN を使用し, 絵を作成する研究 [5] 等がある.

また, 日本で古くから親しまれている芸術として俳句がある. 俳句は世界最短の定型詩とされ, 日本の伝統芸術である. 俳句の特徴として,17 音である, 季語を含んでいる, 切れ字を 1 つ以下含むといったルールが明確に決められている.[1][2] そのため, 他の芸術よりも俳句はルールを使用することにより生成された句の評価が比較的行い易い. 数値的に評価することがたやすく工学的な分析が容易であるため, 深層学習に使用される損失関数の計算を定量的に行うことができる. また, 俳句は長い歴史があり日本で広く親しまれているため, 深層学習で使用される教師データが数多く存在している. 従って, 本研究では芸術の生成に深層学習を使用する一例として俳句を使用し, その有用性を検証する. 俳句や和歌を自動生成する研究は昔からされてきているが, 多くは単語を組み合わせ, 何らかの方法で評価値を出して評価が高いものを出力するという仕組みになっている.[6][8] 中にはディープラーニングを使用したものもあるが, 画像に基づいた俳句を生成するという試みは少ない.[7]

1.2 本研究の目的

本研究では, 深層学習の芸術作成での有用性を示すとともに, 日本の伝統芸術である俳句の作成支援システムの作成を行う. 具体的な俳句の作成方法として, 吟行をはじめとして俳句作成において一般的な方法である「モチーフから

[1] 北海道大学 大学院 情報科学研究科
[a)] yoneda@complex.ist.hokudai.ac.jp
[b)] yokoyama@complex.ist.hokudai.ac.jp
[c)] tomohisa@complex.ist.hokudai.ac.jp
[d)] kawamura@complex.ist.hokudai.ac.jp

俳句をつくる」ことに着目する. 中でもモチーフとして写真を取り上げ, 写真から俳句を作成することを俳句作成過程として, それを深層学習を用いて再現する. 深層学習が芸術作成に有用であるかを検証するため, 作成した俳句に独創性があるか, 俳句のルールを学習できているかといった面から検証を行う. また, 俳句作成において重要なプロセスである「選句」を深層学習を用いて再現できるか確認する.

2. 関連研究

ここでは本研究で使用している技術について解説を行う.

2.1 長・短期記憶

再帰型ニューラルネットワークの勾配消失問題を踏まえて, 長期にわたる記憶を実現できる方法がいくつか提案されている. その中で広く用いられている方法が長・短期記憶 (Long Short-Term Memory, 以下 LSTM とする) である.LSTM では, 上記の基本的な再帰型ニューラルネットワークに対し, その中間層の各ユニットを図 1 のようなメモリセルと呼ばれる要素で置き換えた構造を持ち, それ以外の構造は変わらない. 図 1 の上矢印はセル状態と呼

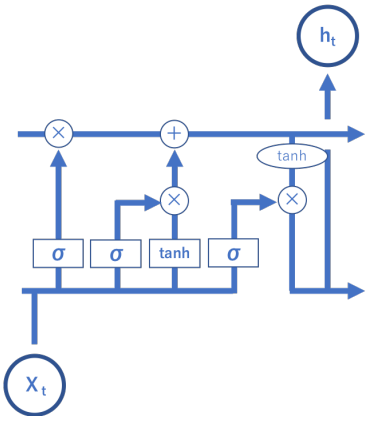


図 1 LSTM のメモリセル概要図

ばれる. メモリセルは 4 つの相互作用する層を持ち, このセル情報に対し情報を削除, 追加する機能を持つ. この操作はゲートと呼ばれる, 選択的に情報を通過させる構造により制御される.4 つの層は 3 つのシグモイド層と 1 つの tanh 層からなっている. 図 1 の左側のゲートから順に説明を行う. まず入力 x_t は忘却ゲートと呼ばれるシグモイド層によってセル情報から捨てる情報を判定する. 数式では $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$ となる. 続いて入力ゲートと呼ばれるシグモイド層でどの値を更新するかを判定する. 数式では $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ と表される. 次に tanh 層ではセル状態に加えられる候補値ベクトル $\tilde{C}$ を作成する. 数式では $\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ で与えられる. 次のステップにおいてはセル情報を更新するためにこれら i_t と $\tilde{C}$ を組み合わせ,$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ を得る. 最後のシグモイド層ではセル状態に基づいて出力を判定するために, セル状態に対して $tanh$ を適用し, それにシグモイド層の出力 o_t を乗算する. 数式ではシグモイド層の出力は $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ と計算され, これを用いて出力 h_t は $h_t = o_t * tanh(C_t)$ となる. こうしたセル状態に記憶を保存する構造により, シンプルな再帰型ニューラルネットワークよりも長期記憶に対して強くなっている.

2.2 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (Convolutional Neurak Network, 以下 CNN とする) は生物の視覚野をに関する神経科学の知見をヒントにした順伝播型ネットワークであり, 隣接層間の特定のユニットのみが結合を持つ特別な層を含んでおり, 主に画像認識に使用される.

CNN は入力側から出力側に向けて, 畳み込み層とプーリング層が順に複数回繰り返されるように並び, その後, 隣接層間のユニットが全結合した全結合層が配置される. 全結合層も一般に複数連続して配置され, 目的がクラス分類ならば最終の出力層はソフトマックス層となる.

3. データ収集

本研究に必要なデータは大きく分けて 3 種類である.

- 俳句データ
- 季語データ
- 俳句と画像のマッチングデータ

以下, それぞれについて解説する.

3.1 俳句データ

俳句を学習する際に使用するデータである. データに含む俳人は多作であり情景を分かりやすく俳句にしているとされる小林一茶, 正岡子規, 高浜虚子の 3 人と, 現代の発想をデータに加えるために現代の俳人多数と大塚凱とした.

まず web 上のデータベース [9] [10] [11] のスクレイピングで収集した. 大塚氏の俳句は本人に提供していただいた. 次に読みがついていない句に対して手動で俳句の読みを付加した. 最後にデータを整理した. 17 音になっていない俳句や季語が含まれない俳句,「(」等の記号を含む俳句は学習データとして不適であるため除外した. 最終的には計 38,506 句となった.

3.2 季語データ

生成した文字列に季語が含まれるかどうかを判定にする際に使用するデータである.

こちらも web 上のデータベース [12] からスクレイピングして収集した. 季語には傍題という, その季語を言い換えた季語が存在しているため, 合せて収集し, 俳句データと同

様に季語の読みを付加した．最終的には 8,665 種となった．

3.3 俳句と画像のマッチングデータ

俳句と画像との適合度を学習する際に使用するデータである．俳句と画像を 1 組として扱うデータであり，このデータで使用する俳句は前述の俳句データで収集した俳句となる．使用する画像は画像素材販売サイト「imagenavi」に提供していただいた画像である．

画像のアノテーションに俳句の季語が含まれていた場合，その画像と俳句を 1 組として扱うことで，機械的に収集した．また，手動で加えることも行った．同様にして俳句と画像の組み合わせの候補を集め，全国のボランティアやアルバイトの方々に，どの画像が俳句に適合しているかを判定して頂き，マッチングデータとして加えた．最終的には 369,754 組収集した．

4. システム

本研究が構築したシステムは，大きく分けて 3 つの構成に分けることができる．

- 俳句生成
- 俳句フィルタ
- 選句

それぞれの部分について解説する．

4.1 俳句生成

3 章で作成した俳句のデータを入力して学習し，学習したモデルを使用して文字列を生成する．ここでは学習部分と生成部分に分けて説明する．

4.1.1 俳句学習

本研究では文章生成において成果を上げている LSTM[13] を使用して俳句を生成する．図 2 に入力データ作成，図 3 に学習のイメージ像を示す．

始めに，学習データをシャッフルし，学習データに使用された文字を出現順に若い番号から ID を与える．次に，学習データの文字をそれぞれの ID に変換し，バッチに分ける．バッチ内のサンプルは，予め設定した BPTT の長さ分の ID の列となる．改行文字にあたる ID が出現した場合でも，そこで区切ることはせず続行する．最後にサンプルを先頭から順に 1-of-K 符号化し projection layer に入力してベクトルに変換 [14] する．そうして得られたベクトルを LSTM に入力する．

LSTM の出力は次に出現する文字の確率であり，正解データは次の文字の ID 番目が 1 のワンホットベクトルである．サンプルを最後まで入力し終えた段階で LSTM の状態をリセットし，誤差を計算する．

パラメータを以下のように設定して学習を行った．LSTM ユニット数は全ての LSTM 層で共通である．本研究では TensorFlow[15] で実装した．

- LSTM 層数:3
- LSTM ユニット数:1024
- 最適化手法:Adam[16]
- 学習率:0.02
- 減衰率:0.99
- エポック数:300
- バッチ数:50
- サンプル長:100
- 目的関数:クロスエントロピー

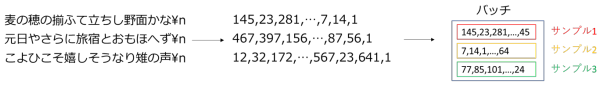


図 2 LSTM の入力データ作成

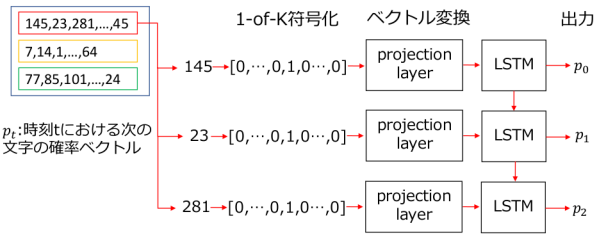


図 3 LSTM の学習

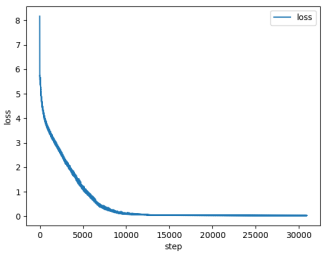


図 4 LSTM の学習の様子

4.1.2 俳句出力

学習した LSTM を使用し，文字列を出力する．図 5 に出力のイメージ像を示す．

入力を改行文字とし，次の文字の確率を計算してルーレット選択をし，選ばれた文字が次の入力となる．改行文字出現してから次の改行文字が出現するまでに選択された文字列が俳句候補となる．改行文字が指定した回数出現した際，出力を停止する．図 6 に生成例を示す．

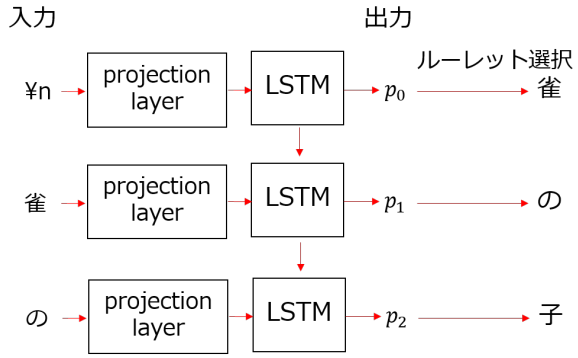


図 5　LSTM の出力

山吹の垣や名月も二度目
眞直にふじまでゆかん冬田哉
水音のふくらんでくる曼珠沙華
寒明の雪どつと来し山家かな
汽車道の鐘におくれて風白し
年忘橙剥いて遊びてり
はつ雪や犬隠過ぎしたりけり

図 6　生成例

4.2　俳句フィルタ

俳句生成部で出力した文字列は俳句としての要素を満たしていない文字列を含んでいるため, 満たしている文字列のみを取り出す. ここでは俳句としての要素を以下のように定義する.

- 17 音になっている
- 季語を 1 つのみ含む
- 切れ字を 1 つ以下含む

本来, 上記の条件を満たしてない句も俳句として認められるが, 句の質が落ちるとされるため, 本研究ではこれらの条件を満たす句のみを俳句とする.

俳句フィルタは生成された文字列に対して上記の条件を満たしているかそれぞれを判定する. 以下にそれぞれの判定方法を述べる. それぞれの判定全てで成功と判定された場合, その文字列を俳句とする.

4.2.1　音数判定

音数判定を行う前準備として, 学習データを使用して漢字とその読み方の辞書を作成する. 漢字の読み方は複数あるものが多いため, 同じ漢字であっても音数が違うことがある. そのため, 音数を正しく測るためにこの処理を行う. 3 章で述べたとおり, 学習データには俳句に対応する読みが含まれているため, 俳句本文と読みで一致する平仮名部分を削除する. そうすることで漢字部分とそれに対応する平仮名部分のみが残る. それらをまとめて記録し, 辞書を作成する.

次に, 生成された文字列に対する音数判定を行う. 文字列に使用されている全ての漢字の読みを前準備として作成した辞書から取り出し, 読みの組み合わせを探索する.17 音となる読みの候補が見つかった場合, 音数判定は成功とする.

4.2.2　季語判定

3 章で作成した季語の辞書から季語を 1 つずつ抜き出し, 生成された文字列にその季語が含まれているかを判定する. 含まれている季語が 1 つのみ存在する場合, 季語判定は成功とする.

4.2.3　切れ字判定

音数判定部で得られた読みの候補を 5 音,7 音,5 音に分け, それぞれの文字列が 「や」「かな」「けり」「なり」で終了している場合, 切れ字が含まれていると判定する. 切れ字が含まれている文字列が 1 つ以下の場合, 切れ字判定は成功とする.

4.3　選句

俳句生成における重要なプロセスとして,「選句」がある. 選句とは, 多くの俳句の中から優れた俳句を選ぶことである. 本研究では, 深層学習を用いて 2 つの方法で選句を再現する.

4.3.1　俳句とモチーフ画像の適合判定

ここでは, フィルタを通して俳句と判定された文字列の中から, モチーフ画像に適合する句を抜き出すことで選句を行う. そのための方法として, 深層学習を使用し画像と俳句がどの程度適合するかを学習する. 適合度が高ければその俳句をモチーフ画像に基づいて生成した俳句であると定める. 学習のイメージ像を図 7 に示す.

前準備として, 使用する学習データに使われている漢字それぞれに出現順に若い番号から ID を与える. 適合判定では,3 章で作成した画像と俳句の組を学習データとして使用する. 負例として, 同数の画像と俳句の実在しない組を学習データと同数だけランダムに生成する.

まず学習データの画像を Inception-v3[17] を使用して特徴量ベクトルを得る. 次に学習データの俳句に使用されている漢字の使用回数を数え, その漢字の ID 番目の要素が使用回数となるベクトルを得る. ここまでで得られた 2 つのベクトルを結合し, 入力ベクトルとする. 適合判定では, 全結合層のニューラルネットワークに入力し, 入力された組が実在の組がどうかを判定する. 学習パラメータを以下のように設定して学習した.

- 中間層:1024,512,256
- 最適化手法:Adam
- 学習率:0.00001
- エポック数:1000
- 目的関数:クロスエントロピー

図 9 に評価例を示す. 評価値が高い俳句とその評価値となっている. この評価値を参考にして選んだ句を俳句の専門家に見て頂いた所, 最後が切れ字で終わる等整ってはい

るが，語の組み合わせに斬新さがないという評価を受けた．そのため，今後は現代特有の語彙を含む俳句を学習するなどの工夫が必要だと考える．

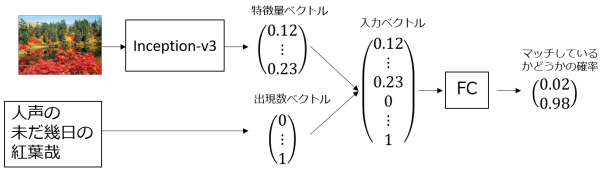


図 7 画像と俳句の適合度学習

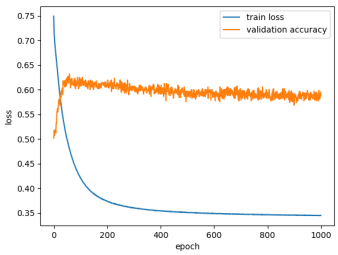
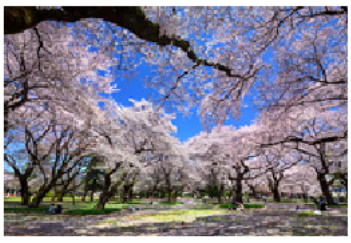


図 8 画像とのマッチング学習



日の本の桜も見えず一つ星,1.0
ものなしはうしろく見えて桜かな,1.0
花見ても肩事もなしに蘆の花,1.0
桜垣濃くらがはれと成にけり,1.0
花の春天窓の窓もあなたかな,1.0
心見や桜がさして子のわられ,1.0
死そはる花のはぐれも花見哉,1.0
小坊主が寝に来て桜咲にけり,1.0
足音の心と並ぶ桜哉,1.0

図 9 画像とのマッチング学習結果

4.3.2 生成句判別

ここでは選句の別の手法として，人間が作成した俳句と 4.1 章で生成した俳句を判別する方法について述べる．

前準備として，3.1 章で収集した現代俳句から 7,911 句を選択し，MeCab で読みを付与し，予め五十音順に設定した ID に変換した．同様にして，本研究で生成した俳句を同数選択し，ID に変換した．

現代俳句を正例，生成した俳句を負例とし，深層学習を使用して 2 値分類した．

使用したモデルは CNN で，設定は以下のとおりである．構造は表 1 に示す．学習の経過を図 10，11 に示す．

表 1 CNN の構造

	入力サイズ	出力サイズ	カーネルサイズ	ストライド
CNN1	17 * 134	17 * 256	3	1
CNN2	17 * 256	17 * 512	3	1
FC1	17 * 512	1 * 256	—	—
FC2	1 * 256	2	—	—

- 最適化手法:Adam
- 学習率:0.00001
- エポック数:500
- バッチ数:100
- 目的関数:クロスエントロピー

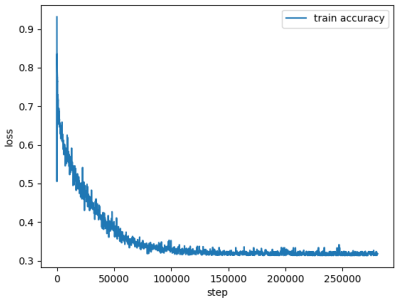


図 10 生成句判別器の訓練誤差

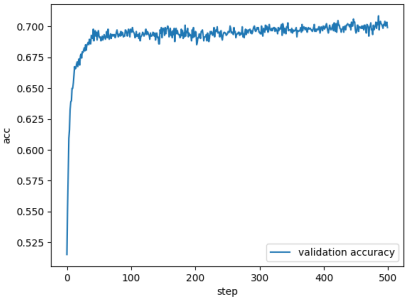


図 11 生成句判別器のテスト正解率

図 12 に評価例を示す. 評価値が高い俳句とその評価値となっている．この評価値を参考にして選んだ句を同様に専門家に見て頂いた所，「鳴き捨てし身のひらひらと木瓜の花」の句のように過去のことを表す「捨てし」と現在のことを表現している「ひらひらと木瓜の花」が混在しているのは矛盾しているという評価を頂いた．そのため，選んだ俳句により良い活用形や語があるかを判断する推敲器の開発が必要だと考える．

黒鳥の光の中に花の雨 1.0
鳴き捨てし身のひらひらと木瓜の花 1.0
猫の子の白夜に人のうしろから 1.0

図 12 生成句判別器学習結果

5. 実験

本研究では 3 つの実験を行う. パラメータを変化させた場合に,LSTM が俳句の性質を学習する能力がどのように変化するかを調べるとともに, 本研究で生成された俳句がモチーフ画像に適合していることを示す.

- 生成された文字列と学習データの類似度に関する実験
- 生成された文字列の俳句フィルタにかけた際の判定の割合に関する実験
- 俳句とモチーフ画像の適合度判定の妥当性に関する実験

以下ではそれぞれの実験について解説する.

5.1 生成された文字列と学習データの類似度に関する実験

この実験では, 生成された文字列と学習に使用した俳句の類似度の最小値を測る実験を行う.

この実験で使用する類似度として,Levenshtein 距離 [18] を使用する. Levenshtein 距離は 2 つの文章の一方を,「挿入」「削除」「置換」の 3 種類の操作でもう一方の文章に変化させる際に必要な最小操作回数である.

以下のパラメータを組み合わせて学習して文字列を 10,000 行生成し, 生成した文字列を学習データとの Levenshtein 距離の最小値をそれぞれ計算した.

- LSTM 層数:2, 3
- LSTM ユニット数:256, 512, 1024

他のパラメータは 4.1.1 章で示したものと同様にした.

図 13, 図 14 に結果を示す.x 軸は最小 Levenshtein 距離,y 軸は該当する俳句の割合であり,l は層数,u はユニット数を表す.

図 13 はどちらもユニット数を 1,024 と設定したものであり, 最小 Levenshtein 割合が高い, つまり学習データと同一の文字列を出力することが多いことが分かる.

図 14 はいずれもユニット数が少ないものであり, 最小 Levenshtein 割合が高く, 学習データにあまり類似していない文字列を出力することが多いことが分かる.

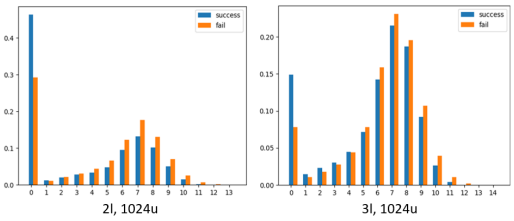


図 13　学習データに類似した分布

5.2 生成された文字列の俳句フィルタにかけた際の判定の割合に関する実験

この実験では生成された文字列と入力データを俳句フィルタにかけて, それぞれの判定の分布から LSTM が俳句として重要である季語や切れ字と言った要素を学習できているかを調べる. 5.1 章と同様のパラメータで学習した複数のモデルでそれぞれ文字列を学習データと同数出力し, フィルタにかけて判定の内訳を見る. 図 15,16,17 に結果を示す.x 軸はそれぞれのモデルであり, 左端は学習データをフィルタにかけた際の判定結果である.y 軸はその条件で該当した俳句数を表す. 図 15 の timeout の項目は, 処理時間削減のために, 音数を探索する処理に 1 秒以上かかる場合に処理を終了した句の数を表す. 図 17 の cannot read の項目は, 切れ字判定に必要な読みを得られなかった句の数を表す.

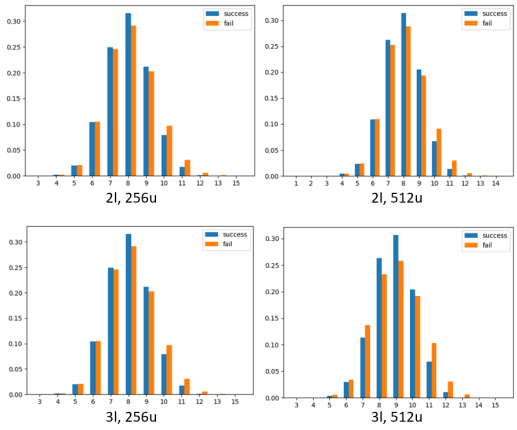


図 14　学習データに類似していない分布

3 つの図全てにおいてユニット数が 1024 のとき, 学習データとの分布に近く, 学習データの俳句に含まれる季語や切れ字といったルールを学習できたといえる.

5.1 章と比較すると, フィルタの判定の分布が学習データに近くなるパラメータは学習データの句を出力する割合が高く, トレードオフの関係であることが分かる. また, 5.1,5.2 章の実験結果を考慮し，本研究では切れ字や季語を含み，かつ独創性のある句を出力しやすいパラメータである 3 層 1024 ユニットが最も良い LSTM のパラメータだと定めた.

5.3 俳句とモチーフ画像の適合度判定の妥当性に関する実験

ここでは, 俳句と画像の適合度判定が妥当であるかを実験する. フィルタを通して得られた俳句と任意の画像を入力していき, 適合度を計算する. 任意の画像に写っているキーワードを設定し, キーワードを含んでいる俳句がそれぞれの評価値でどの程度出現するか確認する. 例えば, 桜の画像なら桜というキーワードである. 結果を図 18,19,20 に示す.

図 18 はキーワードを含む句が多く高い評価をされているが, 図 19,20 は高い評価をされている句が少ない.

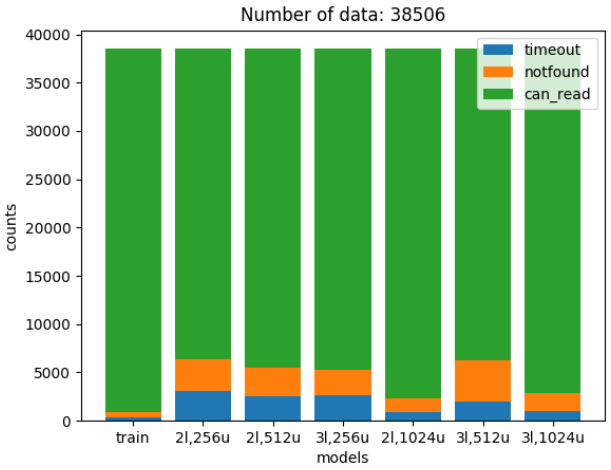


図 15　フィルタ判定分布:音数

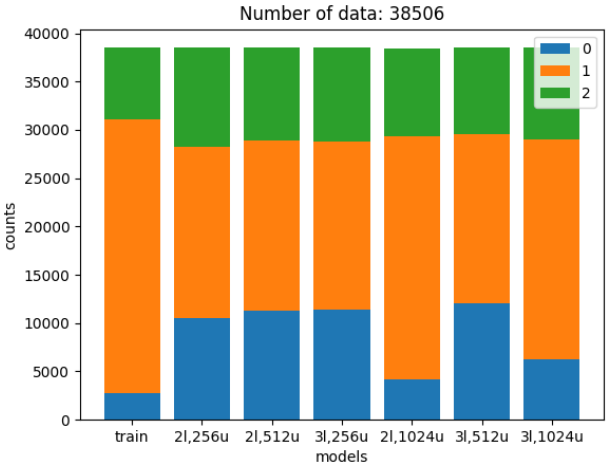


図 16　フィルタ判定分布:季語

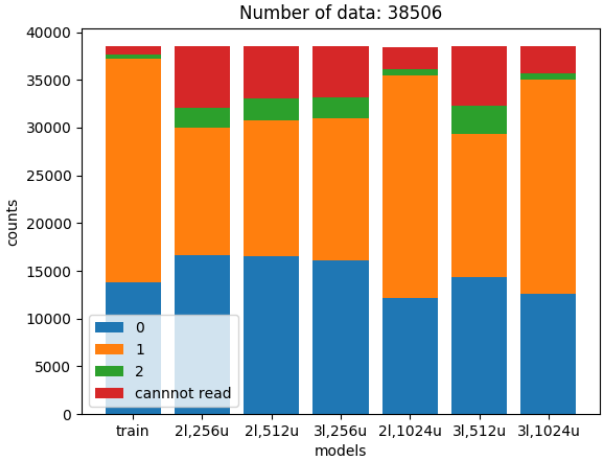


図 17　フィルタ判定分布:切れ字

従って, 画像に適合する俳句を選ぶことができる画像もあるが, 精度は高くないことが分かる.

6. 結論

本研究では一茶など過去の有名な俳人と現在の俳人の句を LSTM に大量に入力することで俳句の音数や季語などのルールを学習した. その学習済み LSTM を使用して文字列を大量に生成し, 俳句としての条件を満たすものを抜き出し, モチーフ画像と適合するかどうか, 学習データに近い句となっているかどうかを算出した.

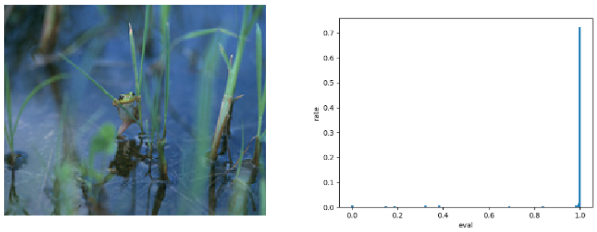


図 18　マッチング評価実験 (キーワード:蛙)

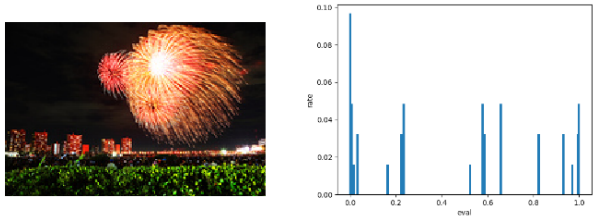


図 19　マッチング評価実験 (キーワード:花火)

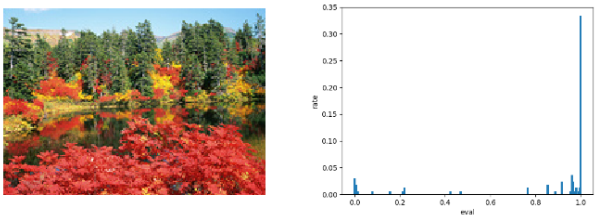


図 20　マッチング評価実験 (キーワード:紅葉)

実験結果より,LSTM を使用することで, 俳句のルールを学習できることを確認した. モチーフ画像と適合する俳句や自然な俳句をある程度抜き出すことができるが, 望ましい句を高く評価できないことも多いため, 今後の課題として選句の評価器の改善が挙げられる. また，選句して得られた俳句にもより適切な副詞や活用形が存在するため，そういった部分を修正する推敲器の開発も課題である.

参考文献

[1] 「新版 20 週俳句入門」 藤田湘子 2010 年
[2] 「超辛口先生の赤ペン俳句教室」 夏井いつき 2014 年
[3] Shujie Liu,Nan Yang,Mu Li,Ming Zhou.”A Recursive Recurrent Neural Network for Statistical Machine Translation”,2014,ACL.
[4] Yonghui Wu, Mike Schuster, Zhifeng Chen, et al.,”Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”,2016,arXiv:1609.08144.
[5] Alec Radford, Luke Metz, Soumith Chintala,”Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”,2015,arXiv.
[6] YANG, Ming, and Masafumi HAGIWARA. ”A Text-based Automatic Waka Generation System using Kan-

sei.” International Journal of Affective Engineering 15.2 (2016): 125-134.
[7] ianchao Wu, Momo Klyen, Kazushige Ito, Zhan Chen ”Haiku Generation Using Deep Neural Networks” 言語処理学会 第 23 回年次大会
[8] Tosa, Naoko, Hideto Obara, and Michihiko Minoh. ”Hitch haiku: An interactive supporting system for composing haiku poem.” International Conference on Entertainment Computing. Springer, Berlin, Heidelberg, 2008.
[9] OPEN Hammerhead 一茶の俳句データベース（一茶俳句全集 V1. 3 0）http://ohh.sisos.co.jp/cgi-bin/openhh/jsearch.cgi?group=hirarajp
[10] 松山市立子規念博物館正岡子規俳句・松山市・CC BY 4.0 http://sikihaku.lesp.co.jp/
[11] 俳句例句データベース http://taka.no.coocan.jp/a5/cgi-bin/HAIKUreikuDB/ZOU.htm
[12] 現代俳句データベース http://www.haiku-data.jp/kigo.html
[13] Sundermeyer, Martin, Ralf Schlter, and Hermann Ney. ”LSTM neural networks for language modeling.” Thirteenth Annual Conference of the International Speech Communication Association. 2012.
[14] MIKOLOV, Tomas, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
[15] TensorFlow https://www.tensorflow.org/
[16] Kingma, Diederik P., and Jimmy Ba. ”Adam: A method for stochastic optimization.” arXiv preprint arXiv:1412.6980 (2014).
[17] SZEGEDY, Christian, et al. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. p. 2818-2826.
[18] Levenshtein, Vladimir I. ”Binary codes capable of correcting deletions, insertions, and reversals.” Soviet physics doklady. Vol. 10. No. 8. 1966.