

おもしろい川柳の生成

太田聖三郎¹ 河原大輔¹ 野村理朗²

¹ 早稲田大学 ² 京都大学

{ota-seizaburo@akane., dkw@}waseda.jp nomura.michio.8u@kyoto-u.ac.jp

概要

本研究では、深層学習による川柳生成モデルの構築を通じて、自然言語処理技術の創造性の向上に焦点を当てる。川柳生成において、音数やお題に沿った内容、おもしろさの学習を組み合わせ、end-to-endでの生成を可能にする。人手評価や自動評価により、提案手法はベースラインに比べて音数制御の精度、川柳のとしての適否、おもしろさの面で優れていることを示した。今後は、人間の川柳との相違点を埋めつつ、モデルの更なる洗練を目指す。

1 はじめに

近年、人工知能による創作活動はめざましい発展を遂げている。自然言語を入力するだけで、その内容に沿った精巧な画像を生成するモデル [1] も登場している。このように、創作モデルの精度が年々向上している中、モデルが人間に取って代わることを危惧する意見も存在する。しかし、これらのモデルは人間の代わりとしてだけではなく、人間の補助としても在り得る。つまり、これからは人間とモデルが共創していく時代である。人間とモデルがうまく協力することでより良い作品の創出が可能となる。

本研究では川柳という文芸に注目し、川柳を生成するモデルを構築する。川柳は俳句などと同様に5-7-5の音数を持ち、お題に対して作るものである。また、皮肉や風刺的な内容を含むことが特徴である。川柳の生成に関する研究は無く、GPT-4などの基盤モデルも依然としておもしろい川柳の生成をすることはできない。本研究では、おもしろい川柳の生成を以下の3つのステップにより達成する。

- 音数・トピックの学習
- 川柳の内容の学習
- 川柳のおもしろさの学習

これらの学習によりモデルに川柳をend-to-endで生成させられることを示した。

2 関連研究

2.1 俳句の生成

川村ら [2] は、LSTM を用いて俳句を生成する手法を提案している。しかし、この手法はモデルがend-to-endで俳句を生成することはできず、俳句生成後にフィルタリングを行っている。また、Transformer [3] 登場以降に俳句や川柳の生成を行った研究は存在していない。本研究では、end-to-endで川柳を生成できるモデルの構築を目標とする。

2.2 PoeLM

Ormazabal ら [4] は教師無し学習による韻律や音韻を学習する手法、Structure-aware training を適用した形式詩生成モデル PoeLM を提案している。この手法では、詩データを必要とせずに学習を行うことができる。川柳は通常の学習コーパスと比べデータ（トークン）数が圧倒的に少ないため、安定した学習を行うことは難しい。しかし、この手法を適用することで川柳以外のデータも使用することができるため、学習の安定化が期待できる。

2.3 LLM における強化学習

昨今の自然言語処理における学習は大規模言語モデル (LLM) に関する学習が主となっている。RLHF (Reinforcement learning from human feedback) [5, 6] はその中でもよく用いられる技術の一つであり、強化学習の報酬モデリングに人間の嗜好を反映させた学習データを用いることで、より人間が好ましいと感じる応答を生成させることを可能にしている。Llama2 [7] では報酬モデリング時に“margin”というペア間の嗜好の差を導入することで報酬モデルの精度を向上させている。本研究では、川柳のおもしろさを人間に評価してもらうことで、モデルがよりおもしろい川柳を生成できるようになることを試みる。

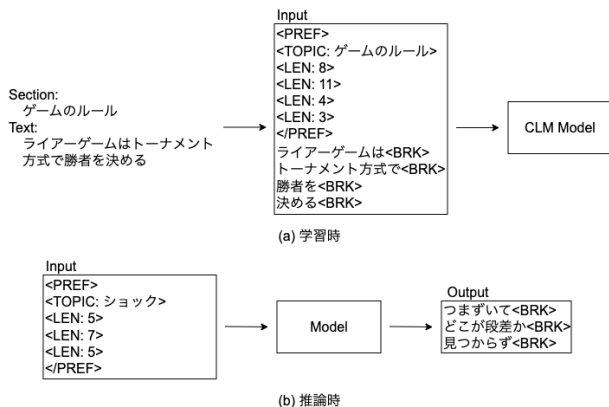


図 1 Structure-aware training

3 提案手法

本節では、言語モデルに良い川柳を end-to-end で生成させるための学習手法を述べる。本研究では 3 つのステップに分けて川柳の特徴を学習する。学習のベースモデルには日本語 GPT-2¹⁾を使用する。

3.1 音数・トピックの学習

通常、言語モデルは音数を明示的に学習することはないが、正確に 5-7-5 の音数で川柳を生成させるためには音数の情報を学習させる必要がある。また、川柳はあるお題に対して作られるものであり、そのお題に沿った内容である必要がある。このお題に対して生成するという構図を明示的に学習することで、川柳を一貫して生成するモデルを構築する。

学習に使用可能な川柳のトークン数は、一般的な言語モデルの学習に使用されるコーパスのトークン数に比べて圧倒的に少ない。そこで 2 節で述べた structure-aware training を用いて、川柳データを使わずに川柳の形式を学習する。PoELM では形式詩の形式を韻律や音韻数の情報を付加することで学習させている。本研究では、川柳の特徴的な要素のうち「5-7-5 の音数を守った生成」、「お題に沿った生成」を目的に学習を進める。図 1 は本研究における structure-aware training の概要図である。

データセット 学習には日本語の Wikipedia データセット²⁾を使用する。このデータセットには記事のタイトル、セクション名、テキストが段落ごとに格納されている。本研究では、セクション名を川柳のお題、テキストを句に見立て、音数の制約やお題に則った生成を行うための学習用に整形を行う。

1) rinna/japanese-gpt2-medium
2) singletongue/wikipedia-utils

川柳における 5-7-5 のようにテキストをある長さで区切り、区切りごとの音数を付与する。また、セクション名を、対応するテキストのトピックとする。本研究では区切りの目安を文節とし、文節の取得には構文解析器 KNP を、音数の取得には形態素解析器 Juman++ から得られる単語の読みを用いる。最終的なデータセットはあるテキストに対するトピック、文節のリスト、文節ごとの音数のリストから成る。データの前処理の詳細は付録 A に示す。最終的なデータ数は 2,282,186 件となる。

モデルの入力は “<PREF><TOPIC: {}><LEN: {}>*n</PREF>({}<BRK>)*n” とし、<PREF>, </PREF>, <BRK>, TOPIC, LEN は特殊トークンとしてトークナイザの語彙に追加する。TOPIC の後には入力文のセクション名、LEN の後には各文節の音数を入力する。n は入力文の文節数である。学習のタスクは CLM (Causal Language Model) で、次のトークンを予測する学習手法である。推論時には <PREF> </PREF> を入力し、川柳の部分を生成する。学習したモデルを “Poelm-ja” と呼ぶ。学習に使用したハイパーパラメータを付録 D に示す。

3.2 川柳の内容の学習

Poelm-ja を川柳生成用に調整する。前節では川柳の形式を学習させたに過ぎないため、川柳を学習データに用いて川柳の内容を学習する。前節では文節を区切りとしていたが、ここでは川柳の 5-7-5 を区切りとすることで、川柳の音数に合わせたファインチューニングを行う。具体的には、3.1 節で適用した structure-aware training を川柳データで行う。

ファインチューニング 川柳投稿サイト「まるせん」より提供された利用可能な川柳データ約 58 万句の内、5-7-5 に分割されていない川柳を 5-7-5 に分割し、全ての川柳を学習に使用できるように前処理を行う。詳細については付録 B に示す。5-7-5 に分割された川柳データを用いて、前節で学習したモデルをファインチューニングする。入力の形式は前節の学習と同様であり、“TOPIC” の後には川柳のお題、“LEN” の後には 5-7-5 を順番に入力する。学習のタスクは CLM である。学習したモデルを “SenryuGPT” と呼ぶ。

3.3 川柳のおもしろさの学習

前節までのステップにより川柳の音数やお題に沿った生成を学習し、さらに、生成する内容を川柳

に整えた。本節では、人間の嗜好をとらえた報酬モデルを使用した強化学習を適用し、よりおもしろい川柳を生成するモデルを構築する。

川柳嗜好データセット Yahoo!クラウドソーシング³⁾を用いて、強化学習時の報酬モデリング用に川柳の良し悪しを比較した川柳嗜好データセットを構築する。あるお題に対する2つの川柳をクラウドワークに提示し、良い（おもしろい）と感じる川柳を選択してもらう。川柳は400個のお題に対し5句ずつ用意し、1つのお題に対して、2句ずつ選んだ総当たりで10ペアを作成する。お題は「まるせん」で開催されるコンテストのお題とChatGPTにより生成したものを使用する。お題ごとに用意する5句の川柳の内、「まるせん」のコンテストの入選作があるお題は1句を入選作とし、4句をSenryuGPTで生成したものとす。入選作がない場合は5句すべてをSenryuGPTで生成する。各ペアに対して10人で評価し、得票数の多かった川柳をお題に対する“chosen”，少なかった川柳を“rejected”と呼ぶ。得票数が等しい場合は、クラウドワークに先に提示された句をchosenとする。また得票数の差に応じて“margin”を設定する。marginの設定の詳細は付録Cに示す。

報酬モデルの構築 川柳嗜好データセットを使用して、お題に対する川柳の良し悪しを回帰により評価する報酬モデルを構築する。ベースモデルにはSenryuBERT [8]を採用する。報酬モデルは、好まれる川柳と好まれない川柳のギャップを最大化するように学習する。損失関数は次式で定式化する。

$$L = -\log(\sigma(R_{chosen} - R_{rejected}) - M) \quad (1)$$

ここで、 σ はシグモイド関数、 R_{chosen} と $R_{rejected}$ はそれぞれ“chosen”，“rejected”の報酬を示す。 M は“margin”である。

強化学習 強化学習にはProximal Policy Optimization (PPO) [9]を使用する。構築した報酬モデルを使用し、SenryuGPTに強化学習を適用する。学習時は、SenryuGPTの入力形式に則って、お題と音数を入力し、川柳部分をモデルに生成させる。それらを元に報酬を最大化するように学習する。学習に使用のお題は「まるせん」のコンテストのお題や、ChatGPTの生成により1,000個収集した。学習に使用したハイパーパラメータを付録Dに示す。学習したモデルを“SenryuGPT-ppo”と呼ぶ。

3) <https://crowdsourcing.yahoo.co.jp/>

4 実験

4.1 実験設定

3節で構築したモデルや既存のモデルによって生成した川柳を評価する。本実験では3つの項目についての評価を行う。

- Structure-aware training による音数制御の精度
- 川柳としての適否
- おもしろさの評価

すべての項目においてクラウドソーシングによる人手評価を行い、音数の精度とおもしろさの評価に関しては自動評価も行う。

ベースライン CLMにより川柳でファインチューニングしたGPT-2 (“Fine-tuned GPT-2”と呼ぶ)をベースラインとする。学習時の入力は“次のお題に対する川柳を生成してください。お題: { }, 川柳: { }”とする。{ }内にはお題や川柳が入る。

モデル 3節で述べた手法の、ステップごとに構築したモデルについて評価を行う。具体的には次のモデルについて実験を行う。

- (a) Poelm-ja (Structure-aware training)
- (b) SenryuGPT (ファインチューニング)
- (c) SenryuGPT-ppo (強化学習)
- (d) Fine-tuned GPT-2 (ベースライン)

以降、モデルは(a), (b), (c), (d)で表す。

実験の詳細 3.2節と3.3節の学習時に使用したお題(seen)のうち50個と、使用していないお題(unseen)50個の合計100個のお題について10句ずつ生成し、それぞれのモデルに対して1,000句を自動評価の対象とする。(a), (b), (c)の生成時は入力を“<PREF><TOPIC: { お題 }><LEN: 5><LEN: 7><LEN: 5></PREF>”とする。サンプリングにより生成する際のtop_pの値は0.9とする。人手評価では、seenとunseenから25個ずつお題を選び、その中から2句ずつ選択して合計100句を評価対象とする。

音数の自動評価では各テキストをKWJA [10]による分かち書きを行った後に、各単語の読みから音数を取得する。おもしろさの自動評価では3.3節で使用した報酬モデルを評価モデルとして使用する。川柳としての適否の自動評価は行わない。音数の測定、川柳としての適否の人手評価は1句あたり5人、おもしろさの評価では1句あたり10人の評価者により評価をする。音数は音数の入力による回答、お

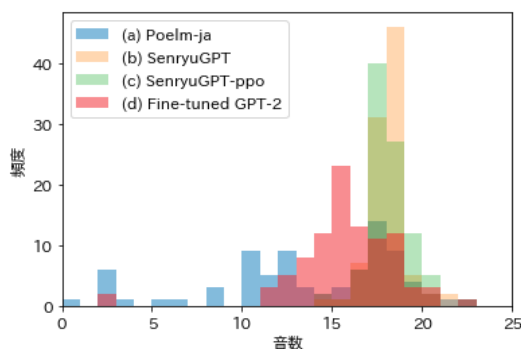


図2 クラウドソーシングによる音数の評価

表1 クラウドソーシングによる川柳としての適否の評価

モデル	カウント
(a) Poelm-ja	4
(b) SenryuGPT	70
(c) SenryuGPT-ppo	71
(d) Fine-tuned GPT-2	18

もしろさは4段階による評価を行う。

4.2 結果

音数の評価 図2にクラウドソーシングによる音数の人手評価結果を示す。KWJAを用いた自動評価結果は人手評価と大きな差異がなかったため、付録Eに示す。(a)は音数の制御を学習しているが、サンプリングにより音数の乱れた生成が多くあることがわかった。一方、川柳を使用して学習した後のモデルはある程度の制御ができており、(d)では17音となっている生成が最も多くなった。ベースラインは15音あたりの生成が多くなり、音数の学習はできていない。また、強化学習が音数の制御に及ぼす影響は見られなかった。

川柳としての適否の評価 表1に、川柳としての適否の人手評価結果を示す。5人の評価者のうち3人以上が「川柳である」と回答したものを川柳とした。(a)は川柳の体裁のみを学習しているため、川柳であるとされた生成は少ないが、川柳で学習したモデルがベースラインと比べ圧倒的に川柳と認められることがわかった。

おもしろさの評価 表2に川柳のおもしろさの人手評価結果を示す。10人の最頻値をその句のスコアとしている。表3に3.3節で使用した報酬モデルによる自動評価結果を示す。(c)の学習時の目的はこの報酬モデルの報酬の最大化であるため、(c)のスコアが突出しているが、強化学習前のモデル(b)もベースラインよりおもしろい川柳を生成できてい

表2 クラウドソーシングによる川柳のおもしろさの評価

モデル	スコア (最頻値)
(b) SenryuGPT	2.202
(c) SenryuGPT-ppo	2.390
(d) Fine-tuned GPT-2	1.701

表3 報酬モデルによる川柳の評価

モデル	スコア (seen)	スコア (unseen)
(a) Poelm-ja	0.0434	0.0370
(b) SenryuGPT	0.0867	0.0694
(c) SenryuGPT-ppo	0.1550	0.1227
(d) Fine-tuned GPT-2	0.0641	0.0493

ることがわかる。また、人手評価の結果から、強化学習により生成川柳がおもしろくなっていることが確認できた。学習で使用したお題については、使用していないお題に比べ、いずれもスコアが高い結果となった。

生成例 以下に各モデルの生成例を示し、講評を行う。“【】”内はトピックやお題を表す。

- 【イベント】その他には ミニマラソンが開かれる (a)
- 【わくわく】うなぎの塔私も挑戦したが (d)
- 【スポーツ】故障したオレはアイススケートで良い (b)
- 【スポーツ】義母からの口撃はいつもブレーキ無し (c)
- 【マイペース】スマホゲームに没頭し過ぎて飛び起きた (c)

1, 2は音数の17音を守っているが川柳と認められなかった例である。(d)は内容が支離滅裂になってしまう場合が多く見られた。3と4の比較では強化学習によるおもしろさの向上が確認できた。ただし、5のように最終的なモデルでも川柳と呼べない生成があることも判明した。

5 おわりに

本研究では、end-to-endで川柳を生成するモデルを構築した。学習を3つのステップに分け、段階を踏んで川柳の特徴を学習していくことで、おもしろい川柳の生成を可能とした。実験の結果、提案手法は音数、川柳としての適否、川柳としておもしろさの3つの項目すべてにおいて、ベースラインを上回る結果となった。ただし、人間の作品とは依然として大きな乖離があり、さらなる研究の余地がある。

謝辞

川柳データを提供いただいた川柳投稿サイト「まるせん」に感謝する。本研究は JSPS 科研費 JP22H01103, JP21H04901 の助成を受けて実施した。

参考文献

- [1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [2] 川村秀憲, 山下倫央, 横山想一郎. 人工知能が俳句を詠む AI 一茶くんの挑戦. オーム社, 2021.
- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.
- [4] Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Findings of the Association for Computational Linguistics: EMNLP 2022**, pp. 3655–3670, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.
- [5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martić, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.
- [6] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 3008–3021. Curran Associates, Inc., 2020.
- [7] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruiti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kam-badur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [8] 太田聖三郎, 河原大輔, 野村理朗. 機械学習を用いた川柳の面白さの予測. 言語処理学会 第 29 回年次大会 発表論文集, 2023.
- [9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [10] 植田暢大, 大村和正, 児玉貴志, 清丸寛一, 村脇有吾, 河原大輔, 黒橋禎夫. Kwja: 汎用言語モデルに基づく日本語解析器. 第 253 回自然言語処理研究会, 京都, 2022.

A Wikipedia データセットの処理

Structure-aware training 用データセットの構築は Wikipedia データセットを基に行う。テキストは段落の一文目を使用し、丸括弧で囲まれている部分を削除する。また、Juman++では英数字の音数が正確に取得できないため英数字を含む文を除外する。そして、長（短）すぎる文は除外する。

B 川柳データの前処理

3.2 節で使用する川柳データの前処理について述べる。「まるせん」より提供された利用可能な川柳データ約 58 万句の内、約 65%は 5-7-5 に分かたれていない。これらの川柳は structure-aware training では適用することができないため、ファインチューニングに使用することができない。この問題を解消するため、あらかじめ 5-7-5 に分かたれている川柳を用いて、川柳 5-7-5 に分割するモデルを構築する。5-7-5 の間の空白を削除した川柳をモデルの入力とし、5-7-5 の間に空白を含んだ川柳を正解として学習を行う。学習に使用したモデルを次に示す。

- rinna/japanese-gpt-2-medium
- rinna/japanese-gpt-neox-3.6b
- megagonlabs/t5-base-japanese-web

GPT 系のモデルでは学習時の入力を“分かたれていない川柳<\s>分かたれた川柳”とし、T5 では分かたれていない川柳の入力に対し、分かたれた川柳をラベルとして学習を行った。3つのモデルの推論結果から多数決を取り、多数派の分割を 5-7-5 の正解と仮定して学習データとする。

C 川柳嗜好データセットの詳細

3.3 節のデータセット構築時に設定した“margin”の設定の詳細を表 4 に示す。得票数の差に応じて設定している。

表 4 得票数の差による margin のアノテーション

比率	margin
10:0	3
9:1	2
8:2	2
7:3	1
6:4	0
5:5	0

D ハイパーパラメータ

表 5, 6, 7 に、3 節の学習で使用したハイパーパラメータを示す。表内の英字は次のモデルに対応している。

- Poelm-ja
- SenryuGPT
- SenryuGPT-ppo
- Fine-tuned GPT-2
- 報酬モデル（強化学習時）

表 5 CLM 学習に使用したハイパーパラメータ

ハイパーパラメータ	(a)	(b)	(d)
learning rate	6e-5	5e-5	5e-5
batch size	128	256	16
#epoch	5	5	2
weight decay	0.2	0	0

表 6 PPO 学習に使用したハイパーパラメータ

ハイパーパラメータ	(c)
learning rate	5e-5
batch size	32
#epoch	5
#ppo epoch	3
clip range	0.1
clipping value	0.2

表 7 報酬モデルの学習に使用したハイパーパラメータ

ハイパーパラメータ	(e)
learning rate	6e-5
batch size	16
gradient accumulation steps	3
#step	5
weight decay	0.5

E KWJA による音数自動評価

図 3 に、KWJA による音数の自動評価結果を示す。人手評価と類似した結果が得られた。

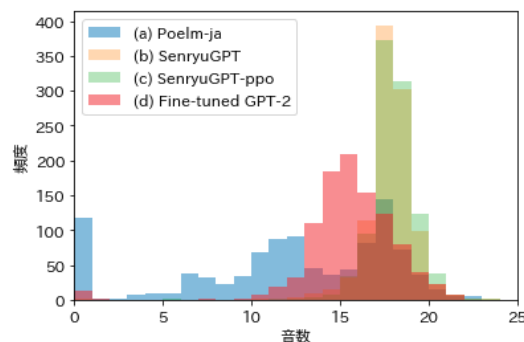


図 3 KWJA による音数の評価