

論文 | Article

品詞別単語使用頻度に基づく
自動作詞ソフトの提案

Proposal of Automatic Lyric Writing Software
Based on the Frequency of
Use of Words for Each Part of Speech

定村 薫

SADAMURA, Kaoru

尚美学園大学
総合政策学部非常勤講師
Shobi University

2024年3月

March.2024

品詞別単語使用頻度に基づく 自動作詞ソフトの提案

定村 薫

Proposal of Automatic Lyric Writing Software Based on the Frequency of Use of Words for Each Part of Speech

SADAMURA, Kaoru

[要旨]

歌詞に特化した新しい自動文章作成ソフトを提案する。

文章作成については、これまでに様々な試みがなされてきた。特に近年では BERT 等の AI モデルの発展や ChatGPT の普及にともない、簡単に文書作成が行えるようになった。本研究は、文書作成を歌詞の作成に特化して、歌詞に含まれる各単語の品詞別にランダムにフレーズを選んでいくという手法によって、従来とは異なる自動作詞ソフトの開発を行った。このソフトでは AI 学習は行っていないが、代表的な自然言語処理モデル BERT を用いた文章のベクトル化による類似度の計算や mlm を用いた校正等を含む。

これまでの文章作成ソフトとの違いは、文章作成用の元になる辞書的なデータとは別にこれから作成したい歌詞のモデルとなる歌詞をもう 1 つ入力する点、それを基に品詞別に単語を選んでいくという点であり、それによって歌詞特有のパターンを表現することができる。

本稿は、この歌詞作成ソフトの概要と自動作詞例について紹介することを目的とする。

[Abstract]

We propose a new automatic sentence creation software specialized in lyrics.

Various attempts have been made to create texts. Especially in recent years, with the development of AI models such as BERT and the spread of ChatGPT, it has become possible to easily create documents. In this research, we specialized in the creation of lyrics and developed automatic lyric writing software that was different from conventional by randomly selecting phrases for each part of speech included in the lyrics. This software does not perform AI learning, but includes calculation of similarity by vectorization of sentences using the representative natural language processing model BERT and proofreading using mlm. The purpose of this paper is to introduce an overview of this lyric creation software and examples of automatic lyric writing.

キーワード

自然言語処理、AI、BERT、歌詞

Keywords:

natural language processing, AI, BERT, lyrics

1. はじめに

人間が普通に使う言語のことを自然言語といい、それに対する分析のことを自然言語処理と呼ぶ。この自然言語処理については、AI研究の発展にともなって、様々な手法が開発されていて、例えばアンケート調査がテキストデータとして得られているような場合の分析やニュース記事の分類等に用いられている。文章作成についても翻訳や要約の作成等については、多くの適用例がみられる。

自然言語処理へのAIの適用については、word2vecに代表される単語の数値ベクトル化、いわゆる単語の分散表現が大きく貢献している。これは単語と単語の類似性を数値ベクトルの距離によって表すという方法で、これによって単語の意味に近いものが数値表現できるようになった。

単一の単語ではなく文章を扱うためにRNN（再帰型ニューラルネットワーク）と呼ばれる仕組みが用いられるようになった。これは文章を時系列のような系列として考え、1語目から順に推定していくというものであり、AIによる翻訳の基礎になっている。その発展形としてLSTMなどの仕組みが次々に発表されてきた。特に2017年に発表されたMulti-Head-Attention機構に基づく分析モデルTransformerと、その発展形としてgoogleが2018年に発表した自然言語処理モデルBERT（Bidirectional Encoder Representations from Transformers）によって自然言語処理は大きな進歩をとげている。BERTでは、すでに事前学習を済ませたモデルを使用できるため、そのモデルを使って個別の仕事（例えば文章分類等）に適用させる（fine tuning）のみで良いので、それまで専門家にしか利用できなかったAIによる問題解決が、多くの時間と労力を費やさずに行えるようになり、AIの普及におおいに貢献した。

自然言語処理の分野では、例えばニュース記事の分類等に上記のBERT等のモデルが使われることが多いが、その適用分野のひとつとして、文章作成があげられる。元々これらのモデルが翻訳に使われていたことから、これは当然の流れと言えよう。特に近年では文章作成についてChatGPTによって対話型の文章生成が行えるようになったことで、誰でも手軽に文章作成が行えるようになってきた。

本研究では、歌詞という形態の文章に特化した新たな歌詞生成ソフトの開発を行った。このソフトの特徴は、品詞に重点を置き、ランダムな語句選択を品詞別に行うという点にある。このソフトではAI学習は行っていないが、代表的な自然言語処理モデルBERTを用いた文章のベクトル化による類似度の計算やmlm（単語の穴埋め）を用いた文章校正等を含む。定村（2019）では、作詞家松本隆の歌詞の特徴を、その使用単語の頻度等によって計量言語学的に分析したが、その過程で使用した単語の分類も利用する。

本稿では、このソフトの概要と作詞例について紹介する。また、いくつかの実行例を示す

ことによって、作成したい歌詞の種類によってデータをどう対応させるか等を考察する。

1. 関連研究

文章作成について、これまでに多くの手法が用いられている。そのいくつかを紹介する。

1.1. マルコフ過程の利用

一般の文章だけでなく、作詞用にも以前からよく使われている手法にマルコフ連鎖の使用を使用した方法があげられる。マルコフ過程とは、ある事象が起きた時、次に起こりうる各事象の確率によって構成されるプロセスである。これを文章作成にあてはめる。文章を時系列的に捉えて、それまでに生成された文の次に来る単語を1語ずつ確率をベースに生成していくという手法である。あらかじめある程度大きな文章データを入力しておき、その中におけるマルコフネットワークを基に、ある単語の次にどの単語が来ることが多いかという確率を計算できるようにしておく。その確率に基づいて文章作成における次の単語の選択を行うという手法である。ただし、元のテキストと似すぎること避けるため、必ずしも最大確率のものを選ぶとは限らない。

すでにマルコフ過程を使用するためのライブラリが提供されているため、ほんの数行のコードで文章生成ができるので、手軽な方法といえるが、欠点としては、確率で選ぶため、元データで使われた事のある単語の組み合わせしか出てこないで、これまでにない斬新なフレーズが出てくることはほとんどない、ということがあげられる。その結果、どこかで聞いたことのあるフレーズが出てきやすい。

1.2. LSTM 等による AI 学習をともなう作成法

RNN、LSTM 等を使って AI 学習を行い、結果として得られたパラメータを用いて、新たな文章を作成するという方法が考えられている。マルコフ法と同様、文章を系列的に捉えて、1語ずつ作成していくという方法である。まず、それまでに生成された文章を数値ベクトル化し、その文章の次に来る語を線形回帰を含む数値的な手法で推測する。つまり、学習用データを用いて、

X: 文章 (数値ベクトル化したもの)

Y: 文章 X の次の語 (数値ベクトル化したもの)

として、X から Y を予測するためのパラメータを推定。これを繰り返し行い学習する。学習後、得られたパラメータを使って、与えられた文章から次の単語を1語ずつ作成していく。

十分な学習を行えば、日本語としては、ある程度実際の歌詞に近い文章が得られるが、マルコフ法と同様の欠点もある。

2. 新たな自動作詞ソフトのアルゴリズム

2.1. 基本的な考え方

これまでに考えられてきた作詞ソフトの多くは、AI を用いて学習を行い、日本語の詞に近づけていくというものだったが、今回開発したソフトは、文章作成の基本部分については AI による学習には頼らず、最後の仕上げの部分にのみ AI モデル (BERT) を使用するという

手法である。

2.2. アルゴリズム

最初にアルゴリズムの全体像をフローチャートで示す。長いため、データ入力と歌詞生成に分けて表示する。データ入力の終了後、歌詞生成が始まる。

図1 データ入力

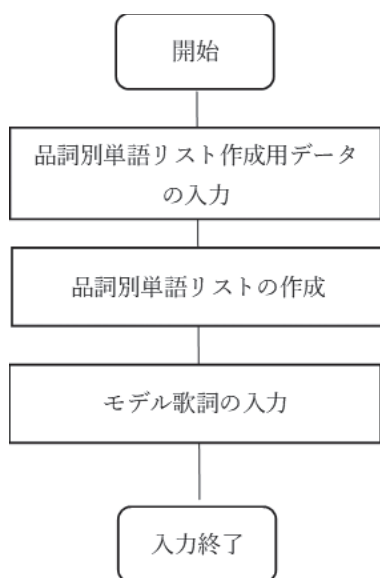
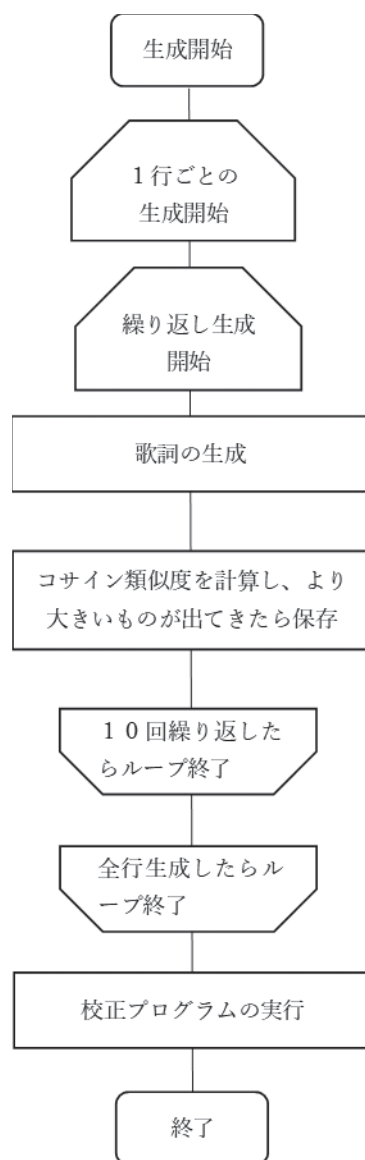


図2 歌詞生成



「品詞別単語リスト作成用データ」「モデル歌詞」という2種類のデータを入力するという点が特徴的であるが、以下にこのアルゴリズムの詳細を説明する。

2.3. 品詞による分類

今回のソフトの大きな特徴は、単語の品詞に注目するという点である。品詞と品詞のつながりによってフレーズが構成される。このことを中心に据えて、日本語として適切なフレーズを作成していく。

2.4. 品詞別単語リストの作成

2.4.1. 歌詞データの準備

今回のソフトでは、作詞の元になるデータは、定村（2019）で分析に使用した歌詞リスト等や歌ネットから必要に応じて歌詞を選び、ソフト向けに若干手を加えたものを利用した。ここではこれを‘品詞別単語リスト作成用データ’と呼ぶことにする。なお、プログラムはpythonで記述した。（プログラムコードは文末にURLを示したgithubから参照可能）以下、わかりやすく説明するために、ここでは特に必要と思われる箇所のみpythonによるコードを示している。

2.4.2. 品詞別単語リストの作成

TEXT ファイルを読み込んだ後、Mecab を使い、

```
m = MeCab.Tagger("-Owakati")
```

テキスト入力後

```
node = m.parseToNode(text)
```

を実行して分かち書きする。次に、品詞別の単語リストを作成する。

例えば、

'色あせたノートに、君の独り言'

というフレーズがあるとき、

```
['色あせ','た','ノート','に','君','の','独り言']
```

をリストにすると、

```
['動詞','助動詞','名詞','助詞','代名詞','助詞','名詞']
```

のようになる

品詞分類は、例えば、形容詞が出てきたら、

```
elif node.feature.split(",")[0] == "形容詞":
```

```
    keiyousi.append(node.feature.split(",")[10])
```

のようなコードにより、keiyousi というリストに単語をひとつずつ加えていく。（前のコードの関係で elif になっている）ただし、動詞については、

```
elif node.feature.split(",")[0] == "動詞" and node.feature.split(",")[5] == "終止形 - 一般":
```

```
    dousi.append(node.feature.split(",")[10])
```

```
elif node.feature.split(",")[0] == "動詞" and node.feature.split(",")[5] != "終止形 - 一般":
```

```
    dousi2.append(node.feature.split(",")[8])
```

のように語尾によっても分類する。（!= は等しくないの意味）

このような手順で、例えば形容詞なら、

['白い','白い','浅い','やさしい','赤い','ほしい','薄い',...]
のようなリストが得られる。

動詞なら、

['会える','振り向く','いく','駆け抜ける','会える',...]]

というようになる。ここで重複している単語があるが、これは後にランダムに単語を選ぶときの確率を多項確率にするためである。

2.5. モデル歌詞の入力と品詞リスト化

多くの作詞ソフトでは、キーワードを入力したり、～～風にといったジャンルを入力するが、今回のソフトでは、モデルとなる4～5行程度の歌詞を1曲選んで、品詞別単語リスト作成用データとは別に入力する。この歌詞をここでは‘モデル歌詞’と呼ぶ。この歌詞から、それらの文の品詞列、すなわち、

['動詞','助動詞','名詞','助詞','代名詞','助詞','名詞']

のようなリストを作成する。これらの品詞ごとに品詞別単語リストからランダムに語句をあてはめていく。

2.6. 歌詞の自動生成

モデル歌詞の品詞リストに従って、文頭から1単語ずつ1.3で作成された品詞別単語リストからランダムに単語を選ぶ。例えばモデル歌詞の1語目が名詞ならば、名詞リストからランダムに単語を抽出する。この際、名詞リストには歌詞作成用ファイルに出現した各単語がその出現回数分登録されているので、度数によって選ばれる確率が決まる。2語目以降も同様にそれぞれの品詞に対応する単語を抽出していく。

このとき、例えば品詞別単語リストに‘風’という単語が5回、光’という単語が1回登録されていれば、‘風’と‘光’の選ばれる確率は5：1になる。ランダム抽出のため、一般には例えば、

病んだところにお前の顔

のような意味のわからない文が生成される事が多い。このため、次節以降に述べるような、様々な改良がなされている。

2.7. 品詞パターンによるフレーズの記憶

単純なランダム抽出だと、上のような意味のない歌詞が生成されてしまうので、これを改善するため、複数の品詞のセットで単語パターンを設定することにした。

例えば、動詞、助動詞、名詞 というパターンについては、品詞リスト作成コードの中に、次のようなコードを入れておく。

```
if node1=="名詞" and node2=="助詞" and node3=="名詞":
```

```
    hins3.append(node3w+node2w+node1w)
```

このパターンが出てきたら、それをリスト hins3 に加えていく。例えば、

時の記憶

のようなフレーズが、これにあたる。このようなパターンを数種類作成する。ただし、作り

たい歌詞のタイプによっては、従来にないフレーズを生成したい場合もあるため、その時は、これらのパターンのうちいくつかを解除して実行することもある。

ここでひとつ問題点が生じる。それは、入力するフレーズによっては、2種類の結合が重なってしまうことがあるということである。例えば、

'名詞','助詞','動詞','助動詞','助動詞'

のような場合、前半の'名詞','助詞','動詞'を結合してしまうと、'動詞'がこれに含まれてしまうため、後半の

'動詞','助動詞','助動詞'

が結合できなくなってしまう。対策としては、モデル歌詞のこの境界部分にコンマ(,)をいれ、品詞分類のときには品詞ではなくそのままコンマとして扱うという方法である。例えば、

'偶然見つけた、引っ越しの夜'

のようにコンマを間に入れておき、品詞リストとしては、

['副詞','動詞','助動詞',' ',' ','名詞','助詞','名詞']

として扱うと、フレーズが'動詞','助動詞'と'名詞','助詞','名詞'に分離できる。

2.8. 単語の分類

定村(2019)に示した単語の分類に倣って、今回のソフトでも単語分類を行う。(付表を参照)例えば同じ名詞であっても、それが'風'や'光'のような自然を表すものなのか、それとも'心'や'夢'のような心情を表したものなのかによって文章が変わってくる。それらを分けるため、品詞別単語リスト作成用ファイルに、光、風などの単語が出てきた場合、名詞ではなくリスト `size` に加える。

```
if node.feature.split(",")[0] == "名詞" and (node.surface=="光" or node.surface=="花"
or node.surface=="風" or node.surface=="空" or node.surface=="雨" or
node.surface=="蒼空" or node.surface=="月" or node.surface=="星"):
    size.append(node.surface)
```

モデル歌詞にそれらの単語が出てきた場合、リスト `size` からランダムに単語を選ぶ。このような分類を名詞や動詞について、何通りか用意した。

2.9. 文末の固定化

歌詞は、男性歌手、女性歌手等によってもパターンが異なる。2.3 に記した品詞別単語リスト作成用データの選定によってある程度単語を選別できるが、単語のランダムな選別だけではうまくいかない部分がある。それが文末である。

例えば女性歌手の場合、「わ」「ね」「の」「よ」が使われることが多い。男性歌手の場合はそこまで極端ではないが、例えば定村(2019)で取り上げた松本隆の作詞曲の場合、「です」「ます」等が多出している。文末にこれらの単語が出てきた場合、「助詞」等の品詞を「わ」「ね」「です」等の単語そのものに置き換え、ランダム化は行わず、そのまま作成された文の末尾につける。

2.10. 品詞別単語リストの更新

歌詞をそのまま使って品詞別単語リストとして使うと、使いにくい単語や元曲のイメージが強すぎて新しい歌詞に使えないフレーズが出てきてしまう。また、特定のフレーズがリフレインによって頻度が過剰になってしまうケースもある。このため、歌詞データから、リフレインを削除する、使えないフレーズを削除する、等の加工が必要になる。

それだけでは十分でない場合は、2.6 で述べた単語の結合等を少なめにした学習用プログラムを作って複数回実行し、作成された文から、作りたい詞にあったフレーズを選んで、品詞別単語リスト作成用データに加えていく。当然、この部分は主観的な要素といえるが、元々作詞は人間が行うものであり、100%ソフトに任せることはできないとの考えから、この部分で人間が介入することはむしろ必要なことであると考えられる。

このようにして品詞別単語リストに更新を重ねた後、再度プログラムを実行して、歌詞を作成する。

2.11. コサイン類似度による文章選択

ここまでの修正で、数回に1回はある程度意味のある文が作成できたので、次の課題は、複数の行に関連性を持たせることである。このため、コサイン類似度を使う。

まず、BERT を使って各行をベクトル化する。その上で2つの行ベクトルの距離の近さをコサイン類似度により数値化する。コサイン類似度とは、2つのベクトルの近さを数値化したものである。

このソフトでは、1行目で歌詞全体のテーマや雰囲気がある程度決定される事を考え、2行目以降の文を各行につき10通り発生させ、それぞれについて1行目との類似度を計算する。次に、モデル歌詞と似たイメージの歌詞である必要があるため、例えば2行目が発生されたとき、モデル歌詞の2行目との類似度も計算する、その2つの値の平均値の最も高い文を10個の文の中から選んでいく。この他、1つ前の行との類似度を判断に加えることも実験的に行ったが、1行目と同じ単語が2行目以降に何度も出てきてしまうという欠点があり、今回は採用しなかった。

以下にこの手順を示す。

① 1行目の作成

10通り作成し、モデル歌詞の1行目に最も類似度が高い文を選ぶ。

② 2行目以降の作成

2行目を10通り作成し、モデル歌詞の2行目との類似度、生成された1行目との類似度の平均値が最も高い文を選ぶ。3行目以降も同様の方法で作成する。

2.12. BERT の mlm による穴埋めを用いた校正

前節の方法で、ある程度近い内容の4~5文が得られるが、これだけでは文の流れ上おかしい単語がどこどころ現れてしまう。また、助詞の使い方が必ずしも適切でなかったりすることもある。これらを解消するために、BERT の穴埋め機能 mlm を用いた。この mlm とは、まず文中の1語を 'MASK' に置き換えておく。これが穴を表すものとして、穴埋めを行っていくというものである。

本プログラムでは、1行ずつ提示させ、その中の修正したい言葉をこちらで選び、これ

を 'MASK' に置き換えて、BERT に穴埋めさせる。その際、その行以前の行との関連性を持たせるため、いったんそこまでの修正済みの文と結合させたいうで、穴埋めを行う。例えば今行っている校正が3行目であれば、すでに校正済みの2行目までの文の後ろに3行目を結合させたいうで3行目の校正を行う。(画面表示上は、単語選択を簡単にするため、該当する行のみを表示するようにしてある) それまでの文も含めて穴埋めを行うことによって、歌詞全体のつながりもある程度得られる。穴埋めのプログラムについては、近江・金田・森長・江間見(2021)に記載のプログラムの関数 `predict_mask_topk` および `beam_search` を使用した。

ここにその具体的な実行例を、次章 3.1 の作成過程において実際に表示された画面を使って説明する。

校正前

階段にゆくよブルース、海に思い出す

↓

校正後

階段には、ブルース、静かに思い出す、

まず修正単語数を3に決める。mlmによってどのように校正されるかわからないので、少し多めに設定しておく場合もある。

まず、「ゆくよ」がおかしいので、変えてみる。次のような画面が表示される。

```
['階段','に','ゆく','よ','ブルース',' ',' ','海','に','思い出す','\n']
```

修正単語数

そこで、修正単語数が3個であれば、3と入力する。次に修正する文字番号を入力する。

1文字目が0なので、「よ」を修正する場合は3となる。

```
['階段','に','ゆく','よ','ブルース',' ',' ','海','に','思い出す','\n']
```

修正単語数 3

文字番号

```
['階段','に','ゆく','[MASK]','ブルース',' ',' ','海','に','思い出す','\n']
```

となった後「よ」が「、」に修正される。次に「ゆく」も変えたいときは、その文字番号2を指定する。

```
['階段','に','ゆく',' ','ブルース',' ',' ','海','に','思い出す','\n']
```

修正単語数 3

文字番号

```
['階段','に','[MASK]',' ','ブルース',' ',' ','海','に','思い出す','\n']
```

となった後「ゆく」が「は」に修正される。

```
['階段','に','は',' ','ブルース',' ',' ','海','に','思い出す','\n']
```

次に「海」が前後の文脈に関連性がないので、その番号6を入力する。

['階段', 'に', 'は', '、', 'ブルース', '、', '海', 'に', '思い出す', '¥n']

修正単語数 3

文字番号

['階段', 'に', 'は', '、', 'ブルース', '、', '[MASK]', 'に', '思い出す', '¥n']

となった後「海」が「静か」に修正される。

この例では、最後の「静か」が想定外の単語で、mlmが思わぬ効果を発揮したケースといえる。

3. 自動作詞ソフトの実行例

3.1. 作詞例 1

モデル歌詞として、織部壮氏の2つの未発表曲の歌詞を、作詞者の許可をいただいた上で使用した。

モデル歌詞 1: 「手の中のハーモニカ」(作詞 織部壮)

胸に水平、消えない夜は
お気に入りのブルース、海に捧げる
けして魂は売らなかったね
ピアノ叩く背中が神様だった
手の中で泣いている、眠れないハーモニカ
・・・

上記の5行を対象とした。

名詞、名詞パターンを避けるため、「お気に入り」は「気に入り」、「神様」は「神」として入力した。

品詞別単語リスト作成用データについて

元々、五木寛之の小説「海を見ていたジョニー」のシーンを思い浮かべて作られた詞なので、横浜を舞台にした柳ジョージ(作詞トシ・スミカワ)などの曲の歌詞を中心に、歌ネット等から雰囲気に近い歌詞を選んで1次歌詞ファイルとし、それから2次ファイルを作成していくという方法で、品詞別単語リスト作成用データを作った。

類似度については、生成された1行目との類似度とモデル歌詞との類似度の平均値を最大にする文を選んでいった。

ソフトにより生成された歌詞の例

夜空には流れ星響いてるピアノは、
階段には、ブルース、静かに思い出す、
音楽には疲れちゃったね、
街角の夜空が、風に。
明日の夜空を迎えに来る、見上げた流れ星

3.2. 作詞例2

モデル歌詞2：「遺言」（作詞 織部壮）

色あせたノートに君の独り言

偶然見つけた引っ越しの夜

洒落た暗号、切なすぎる（刹那すぎる）

抱きしめたい、抱きしめたい

時の砂漠で泣き崩れる

抱きしめたい、抱きしめたい

灰になっても、風になっても

・・・

「抱きしめたい」のリフレインは自動作詞に適さないため、それを除く5行を対象とした。また、「切なすぎる」は‘形容詞“動詞’のパターンだが、このパターンについて品詞別単語リストの度数が少なすぎてうまくいかなかったため、元々ダブルミーニングになっている「刹那すぎる」のほうを使った。類似度の対象については、3.1と同様である。品詞別単語リスト作成用データは、松本隆の歌詞を中心に、モデル歌詞にイメージの近いものを集めて作った。

ソフトにより生成された歌詞の例

残したものはいつも遠く、

ずっと出ない、東京の夜空、

落っこちた世界、ほら。。

夏の季節を駆け抜ける、

どこに行っても、風に溶けて

3行目の「ほら」は mlm 校正で偶然出てきたものである。（その次の単語は句点に修正されている）

これらの2つの例を見ると、品詞別単語リスト作成用データによって詞の内容がかなり影響されることがわかる。

3.3. 小説を品詞別単語リスト作成用データとして使用する

ここまで品詞別単語リスト作成用データとして歌詞を使用してきたが、大量のデータを集めるのが難しいため、繰り返し作詞させると、データ数不足により同じようなフレーズが作成されやすくなってしまふ。文字数の多い中長編小説を使用すれば、この問題はある程度解決する。ただし、小説のフレーズが歌詞になじむかという問題はある。このため様々な小説についていくつかのモデル歌詞との相性を探ってみたところ、次の組み合わせがうまく適合した。

小説データ：「銀河鉄道の夜」 宮沢賢治

モデル歌詞：「12月の雨の日」はっぴいえんど（1971年、作詞 松本隆）

「銀河鉄道の夜」については青空文庫よりダウンロードしたものを使用した。2.9に示した単語リストの更新は最小限（「ジョバンニ」「カムパネルラ」をはずす等）しか行っていない。

「12月の雨の日」の各行の品詞リストは、次のようになる。ただし「自然」「街」は本稿で定義したものである。（付表を参照）

['名詞', '助詞', '名詞', '助詞', '形容詞', '名詞', '助詞']

['自然', '助詞', '動詞', '助動詞', '名詞', '助詞', '動詞']

['名詞', '助詞', '街', '助詞', '自然', '助詞', '名詞', '助動詞', '動詞']

['動詞', '名詞', '助詞', '代名詞', '助詞', '動詞', '助詞', '動詞']

ソフトにより生成された歌詞の例

水のなかに流れる音に、
星に似た家がある、
月夜の町に星がいっぱいになる、
かがやく三角を私は見ました

ソフトによる若干の mlm 校正は加えられているものの、このレベルの歌詞が4～5回に1回程度出力される。また、宮沢賢治らしさも所々出ている。このように、品詞リスト作成用データとモデル歌詞の相性が良ければ、このような歌詞を簡単に作成できる。

4. まとめと今後の課題

4.1. 人間の感覚をどのように反映するか

近年ブームになっている ChatGPT を使うと、文章を簡単に作成できるが、これは主に報告書など事実をまとめるという分野において最も効果を発揮する。しかし、小説や歌詞のような人間の感覚によって評価される種類の文章については、必ずしも使用者の思ったとおりのものが作成されるとは限らない。この部分については、どうしても個人の評価を何らかの形で入力していく必要がある。

本稿のプログラムでも、2.9に示した品詞別単語リストの再加工で、使えるフレーズ、使えないフレーズの判別を人間が行っている。汎用性を重視するならば、できるだけサイズの大きい品詞別単語リストが必要である。しかし、サイズを大きくすればするほど、全く必要のないフレーズが紛れ込んでくるリスクも大きくなってしまふ。これを避けるためにも、ある程度選別されたテキストファイルを用意する必要がある。つまり unnecessary フレーズを削除し、必要なフレーズを加えていく等の2次加工が必要となる。

また、BERT の mlm による校正において、どの単語を置き換えるかについては個人の判断も必要になっている。

4.2. 各行の関連性について

本稿では、4～5行程度の歌詞を生成しているが、一番問題となるのはそれらをまとめた1つの歌詞としてとらえたとき、各文のつながりや流れがうまくいっているかということである。類似度の計算により、ある程度関連のある文を選ぶことはできたが、それだけでうまくいくわけではない。また、生成後の校正をBERTのmlmによって行うことによって、対象となる単語の前文と後文のつながりを考慮した校正が行えるが、必ずしも思うような単語が出てくるとは限らない。また、校正する単語の順序によって結果が変わってくるため、場合によってはmlmによる校正を繰り返し試す必要がある。

また、類似度の高さによる文の選択を行うと、場合によっては同じ単語が各行に重複してしまいやすいという欠点もある。

これらの点については、今後さらに検討していきたい。

4.3. ソフト作詞に向く詞のタイプ

今回、いくつかのモデル歌詞を用いて作詞を試みってみたが、その結果わかったことは、「時」とか「心」などのような抽象的な単語を使った比喩的表現は比較的作りやすい。ところが、定村（2019）で取り上げた松本隆の女性歌手向けの曲のような、具体的な花の名前等が多く含まれる歌詞、すなわち情景描写を多く含む歌詞は、現状ではうまく作れていないということである。これは具体的な名詞を多く使えば使うほど、他の名詞や動詞との組み合わせも膨大になり、使えないフレーズが出続けるためである。

例えば「時が消した」や「時は忍び足で」といったフレーズであれば擬人法として成り立つが、花の名前、例えば「ひまわり」や「スイートピー」で同様のことを行うと、60年代のサイケソングでもない限り成り立たないフレーズが出やすくなってしまふ。逆に制約をつけすぎて「赤いバラ」のような当たり前のフレーズしか作成できないということになってしまふのも問題である。

本稿のために行った実験的な歌詞制作においても、情景描写を少なめにした品詞リスト作成用データをもとに作成された歌詞のほうがうまくいっている。情景描写をどうするかが大きな課題である。

3.3に示した「銀河鉄道の夜」がある程度うまくいったことから、他の小説を使った歌詞生成も試みる価値はあるかもしれない。ただし、青空文庫に収録されている小説は著作権の切れた古典ばかりなので、現代的な歌詞を生成できるものを探すのは難しいと思われる。

5. おわりに

歌詞の自動作成という点では一定の成果が得られたといえる。ただ、結果として膨大な行数のプログラムになってしまった。AIをうまく使えば、これが数分の1の行数でできる可能性もある。つまり、本ソフトで用いた作成過程をうまく盛り込んだ手順で繰り返し学習する方法があれば、真の意味での自動歌詞作成ソフトになると思う。

100% AIに頼って学習させるのか、それとも本稿のように人間が様々な要素を加えていくのか。当然、人間が介入したほうが良い作品ができるはずだが、自動化という観点からは、できるだけ人間の労力を軽減する手法のほうが優れているといえる。その意味で、人間の判

断をうまく取り入れた学習を自動で行う AI が出来れば理想的である。しかし、小説や歌詞のような人間の感情に訴える文章については、全自動化には限界があるように思える。ソフトは単に良いフレーズや文を作るためのものであり、その後に人間がそれらを組み合わせたり修正したりしながら完成させれば良いという考え方もできる。

こういった人間と AI の共存については、今後考えていかなければならない課題であろう。

付表：単語分類表

名詞

自然	光、花、風、空、蒼空、雨、月、星
街	街、町
心情	思い出、想い出、夢、心、こころ、気持ち
花	桜、さくら、バラ、薔薇、すみれ、ひまわり、向日葵、フリージア、たんぽぽ、チューリップ、あじさい、紫陽花、オレンジ、みかん、ミカン、いちご、苺、カトレア、カーネーション、'ゆり、ユリ、百合、コスモス、シクラメン、ジャスミン、すずらん、スズラン、れんげ、林檎、リンゴ、レモン、檸檬、パイナップル、ブルーベリー、blueberry、月見草
季節	春、夏、秋、冬
色	白、黒、赤、青、黄色、紫、緑、グレー、灰色
乗り物	電車、汽車、列車、車、自転車
場所	海、山、川、丘、岬、高原、避暑地
日	昨日、今日、明日

動詞

恋愛	愛する、恋する、好き、恋し、愛し
行動	歩く、走る、来る、行く
見る聞く言う	見る、聞く、聴く、言う

参考文献

- [1] 定村薫 (2019) 尚美学園大学総合政策研究紀要,34,17-33 (2019-09)
- [2] 近江崇宏・金田健太郎・森長誠・江間見亜利 (2021) 「BERT による自然言語処理入門」、オーム社
- [3] 杉山聡 (2022) 「本質を捉えたデータ分析のための分析モデル入門」 ソシム株式会社
- [4] 樋口耕一 (2014) 「社会調査のための計量テキスト分析」 ナカニシヤ出版
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (2017). *Attention Is All You Need*. In *Advances in Neural Information Processing Systems*, pages 6000–6010.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). *BERT: pre-training of deep bidirectional transformers for language understanding*. *CoRR*, Vol. abs/1810.04805, ..

歌ネット (<https://www.uta-net.com/>)

青空文庫 (<https://www.aozora.gr.jp/>)

プログラムコード (GitHub)

<https://github.com/kyomu0/lyric/blob/main/lyric.ipynb>

(注) GPU 使用、品詞別単語リスト作成用データとして UTF-8 エンコードの TEXT ファイルを用意すれば実行可能。

