

# 分散的意味表現を利用した自動作詞

Lyrics generator exploiting distributed semantic representation

堀 玄<sup>\*1\*2</sup>      嵯峨山 茂樹<sup>\*3</sup>  
Gen Hori      Shigeki Sagayama

<sup>\*1</sup>亜細亜大学      <sup>\*2</sup>理化学研究所      <sup>\*3</sup>明治大学  
Asia University      RIKEN      Meiji University

We develop a Japanese lyrics generator system for use with a lyrics-based automatic composition system such as *Orpheus*. By modeling the character of lyrics corpus using a bigram model modified with distributed semantic representation and formulating lyrics generation as a path finding on the vocabulary set, we realize a lyrics generator that respects user's intention through user-designated keywords and complies with the length limitation. The lyrics generator system has been implemented as a web application and open to the public.

## 1. はじめに

近年の人工知能に対する社会的関心にとともに、人工知能による芸術創作にも注目が集まるようになった。また歌声合成ソフトウェアの普及を背景としてアマチュアによる自作曲発表が急増し、作詞・作曲を支援する技術の需要も大きくなっていく。日本語の自動作詞に関する先行研究としては[樺島 70], [伊藤 98], [阿部 11] が挙げられる。これらはいずれもバイグラムモデルまたはトライグラムモデルに基づき文頭から文末へと語彙を配列していくものであり、まず歌詞の先頭部分を定めないと残りの部分の生成ができない。これに対して本研究では、歌詞生成を語彙リスト上の経路探索問題として定式化することにより、歌詞の先頭部分を指定する必要がなく、作詞意図を示す少数のキーワードを指定するだけで歌詞生成できる原理を利用する。また、バイグラムモデルまたはトライグラムモデルに基づく自動作詞には、歌詞コーパスのもつ情報のうち2語または3語以上離れた遠距離の単語間の関連性が反映されないという欠点があるが、本研究では、語彙の分散的意味表現を利用してバイグラムモデルを修正することにより、歌詞コーパスのもつ遠距離の単語間の関連性を反映した歌詞を生成する方法を提案する。本研究の自動作詞システムは、歌詞の韻律に基づく自動作曲システム[嵯峨山 09]への入力を想定して開発しており、Webアプリケーションとして実装し自動作詞作曲システム *Orpheus* として公開している。

## 2. 歌詞の確率的言語モデル

### 2.1 バイグラムモデル

文中の単語の出現確率がその直前の  $(N - 1)$  単語だけに依存すると仮定する確率的言語モデルを  $N$  グラムモデルと呼ぶ。特に  $N = 2$  の場合をバイグラムモデル、 $N = 3$  の場合をトライグラムモデルと呼ぶ。バイグラムモデルは単語の出現確率が直前の1単語だけに依存すると仮定するモデルであり、遷移確率  $a_{ij}$  (単語  $w_i$  の次に単語  $w_j$  が出現する確率) を与えると定まる。本研究では、バイグラムモデルをベースとして、遷移確率に対して次節で述べる分散的意味表現に基づく修正を行った確率的言語モデルを利用して、ユーザが入力したキーワードを含む歌詞を生成する。

バイグラムモデルを推定するために、まず歌詞コーパス<sup>\*1</sup>の各行を形態素解析<sup>\*2</sup>する。次に形態素解析の結果から形態素間の遷移回数を数え上げ、これをもとに相対頻度として遷移確率の最尤推定値を求める。この際、各行の形態素解析の結果の最初と最後に、文頭と文末を示す特殊記号 BOS と EOS を加えて遷移確率の推定を行う。さらに、歌詞コーパスに現れなかった遷移の確率がゼロになる問題(ゼロ頻度問題)を回避するため、グッド・チューリング法による遷移確率の平滑化(スムージング)を行う。

### 2.2 分散的意味表現

バイグラムモデルに基づき、遷移確率だけを利用して歌詞の生成を行うと、歌詞コーパスのもつ情報のうち2語以上離れた遠距離の単語間の関連性が反映されなくなる。日本語では内容語どうしの中に助詞をはさむ構文が基本的であるため、何らかの方法で遠距離の関連性を反映させることが必要となる。そこで本研究では、単語の分散的意味表現を利用して、生成される歌詞に遠距離の関連性を反映させる方法を提案する。

分散的意味表現は(1-of-N 表現などと比べると)低次元のベクトル空間に単語を埋め込む手法であり、意味的に近い単語はベクトル空間においても近いベクトルとして表現される。[Mikolov 13]による word2vec は、コーパスから単語の分散的意味表現を学習するアルゴリズムであり、得られた分散的意味表現を自然言語処理の各種タスクに適用すると従来手法と比較して高い性能を得られることが報告されている。単語  $w_i, w_j$  の表現ベクトルを  $\mathbf{w}_i, \mathbf{w}_j$  とすると、両者が意味的に近い場合、コサイン類似度  $\mathbf{w}_i \cdot \mathbf{w}_j / \sqrt{\|\mathbf{w}_i\| \cdot \|\mathbf{w}_j\|}$  の値が大きくなる。この点から、表現ベクトルはノルムではなくその方向が意味をになっており、実質的に単位超球面上の点と同一視できると考えられる。以降では各単語  $w_i$  の表現ベクトル  $\mathbf{w}_i$  は単位ベクトルに規格化されているものとする。この場合、コサイン類似度は  $\mathbf{w}_i \cdot \mathbf{w}_j$  と表される。また、単語  $w_i, w_j$  が意味的に近い場合、表現ベクトル  $\mathbf{w}_i, \mathbf{w}_j$  は単位超球面上で近い点に対応する。

本研究では、歌詞コーパスを学習データとして word2vec による学習を行い、各単語に対応する 200 次元の表現ベクトルを得た。学習した表現ベクトルどうしのコサイン類似度の分布

\*1 これまでに紅白歌合戦で歌われた楽曲約 3000 曲の歌詞からなる歌詞コーパスを用いた。

\*2 奈良先端科学技術大学院大学で開発された形態素解析エンジン「茶筌」を用いた。

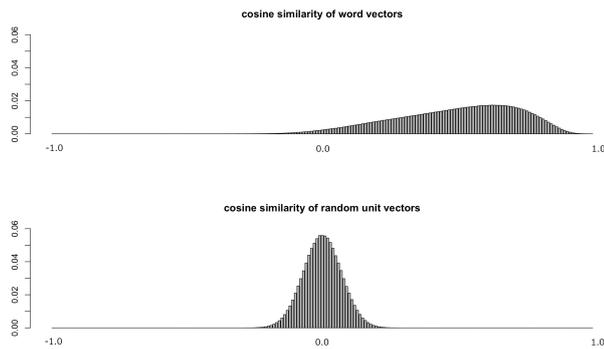


図 1: 歌詞コーパスから学習した 200 次元表現ベクトルどうしのコサイン類似度の分布 (上) と, ランダムな 200 次元単位ベクトルどうしのコサイン類似度の分布 (下)

は図 1 (上) のようになった。一方, 図 1 (下) はランダムな 200 次元単位ベクトルどうしのコサイン類似度の分布である。単位超球面上に一樣に分布する単位ベクトルどうしのコサイン類似度は, 次元が上がると平均付近に集中した分布になる。両者の比較より, 実際にコーパスから学習した表現ベクトルは単位超球面上で特定の領域に偏っていると考えられる。

コーパスのもつ遠距離の単語間の関連性を反映させるため, ユーザが入力したキーワードと遷移先の単語のコサイン類似度が大きいほど遷移確率が大きくなるような修正を行う。具体的には, 前節で求めた遷移確率  $a_{ij}$  を次のように修正する。

$$\frac{1}{C_i} a_{ij} \cdot \exp(\lambda(\bar{\mathbf{k}} \cdot \mathbf{w}_j - 1))$$

ここで,  $C_i = \sum_j a_{ij} \cdot \exp(\lambda(\bar{\mathbf{k}} \cdot \mathbf{w}_j - 1))$  は遷移先に関する確率の合計を 1 とするための定数,  $\lambda$  は修正の強さを表す定数,  $\bar{\mathbf{k}}$  はユーザが入力したキーワード  $k_1, \dots, k_n$  の表現ベクトル  $\mathbf{k}_1, \dots, \mathbf{k}_n$  の重心を規格化したものである。ユーザが入力したキーワードとの類似度が大きい単語の出現確率を大きくすることにより, キーワードを介して遠距離の単語間の関連性を反映させる。

### 3. 動的計画法による歌詞の生成

歌詞の生成は, 歌詞コーパスを形態素解析して得られた形態素リストから適切なものを選択し配列することにより行う。これは図 2 のような形態素リスト上の経路探索と考えられる。図 2 の左側は歌詞コーパスを形態素解析して得られた形態素のリストであり, このリスト上で, BOS から始まりユーザが入力したキーワード  $k_1, \dots, k_n$  を通過し EOS で終わる経路が 1 行の歌詞を与える。前節で導入した歌詞の確率的言語モデルに基づき, 確率が最大となる経路を探索することにより, 所定のキーワードを含み歌詞コーパスの単語配列の特徴を保つ歌詞を生成することができる。このような経路探索問題は, 動的計画法 (DP) のアルゴリズムで効率的に解くことができる。特にキーワード  $k_1, \dots, k_n$  を通過する経路の探索問題は, [迫江 71] による 2 段 DP に基づく次のような手順で解くことができる。第 1 段階では, ユーザが入力したキーワード  $k_1, \dots, k_n$  に便宜的に  $k_0 = \text{BOS}$  と  $k_{n+1} = \text{EOS}$  をつけ加え,  $k_{i-1}$  と  $k_i$  をつなぐ各経路 (第  $i$  区間と呼ぶ) を順に探索する。この段階では経路上の形態素数にある程度の幅をもたせ, 各区間について経路の候補を複数求める。また計算時間短縮のためにビームサーチを行う。第 2 段階では, 各区間について求めた候補を総当た

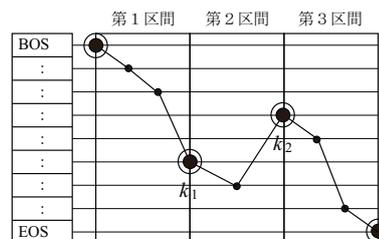


図 2: 形態素リスト上の経路探索による歌詞生成の原理: BOS から始まりキーワード  $k_1, \dots, k_n$  を通過し EOS で終わる経路が 1 行の歌詞に対応する。

りで組み合わせて 1 行の歌詞とし, ユーザが指定した文字数からのずれが一定の値より小さいものを最終的な候補とする。最終的な候補を確率が大きい順にソートし, 上位から所定の数を表示する。

### 4. 自動作詞システムの実装

以上の歌詞生成の原理に基づく自動作詞システムを Web アプリケーションとして実装し, 自動作詞作曲システム *Orpheus*<sup>\*3</sup> の一部として公開している。このシステムでは, ユーザは 5 個までのキーワード, 歌詞の 1 行の文字数, 歌詞の行数を指定して作詞を行うことができる。ユーザが入力したキーワードは形態素解析され, 名詞, 動詞, 形容詞だけが使われる。経路探索の結果から確率が大きい順に 10 個の候補が表示され, 選択した歌詞は自動作曲システムの歌詞入力欄に転送できる。

### 5. おわりに

本研究では, 分散の意味表現を利用して修正したバイグラムモデルと経路探索により, ユーザが入力したキーワードを含む歌詞を生成する方法を提案した。これは, バイグラム確率あるいはトライグラム確率に基づき, 文頭から文末へと語彙を配列する先行研究の方法とは原理的に異なる方法である。今後の研究課題としては, 分散の意味表現を利用して複数行にわたって意味的なつながりのある歌詞を生成するアルゴリズムの開発が挙げられる。

### 参考文献

[樺島 70] 樺島 忠夫: 流行歌をつくる ～国語学・国語表現法資料～, 計量国語学, Vol. 52, pp.8–40 (1970)

[伊藤 98] 伊藤 雅光: ユーミンの言語学 (17), 日本語学, Vol. 17, No. 9, pp. 92–97 (1998)

[阿部 11] 阿部 ちひろ, 伊藤 彰則: 統計的言語モデルを用いた作詞補助システム, 情報処理学会研究報告, 2011-MUS-91(9) (2011)

[嵯峨山 09] 嵯峨山 茂樹, 中妻 啓, 深山 覚, 酒向 慎司, 西本 卓也: 日本語歌詞からの自動作曲, オペレーションズ・リサーチ: 経営の科学, Vol. 54, pp. 546–553 (2009)

[迫江 71] 迫江 博昭, 千葉 成美: 動的計画法を利用した音声の時間正規化に基づく連続単語認識, 日本音響学会誌, Vol. 27, No. 9, pp. 483–490 (1971)

[Mikolov 13] Mikolov, T., Chen, K., Corrado, G., and Dean, J.: Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013)

\*3 <http://www.orpheus-music.org/v3/>