ありがちでない歌詞生成に向けた 曲調と歌詞の関係に基づくベクトル空間モデル

渡邉 研斗　後藤 真孝
産業技術総合研究所 (AIST)
{kento.watanabe, m.goto}@aist.go.jp

1 はじめに

歌詞情報処理の主要技術に自動歌詞生成がある.従来の自動歌詞生成に関する研究では,メロディに対して歌いやすい歌詞や [1,2],押韻を考慮した歌詞など [3],歌詞特有の性質に焦点が当てられてきた.しかし,これらは歌詞特有の性質の一部に過ぎず,多様な歌詞を生成するためには,考慮すべき性質の拡充が重要となる.本研究では,自動歌詞生成の可能性を広げる上で,以下の性質に着目した:

- バラード楽曲では “love” が,メタル楽曲では “kill” が使われやすいように,曲調に歌詞をあわせることがある.しかし,従来手法には,曲調を考慮した自動歌詞生成技術はない.
- 作詞作業では,ありがちな表現とありがちでない表現を使い分けることで歌詞を作成するため,自動歌詞生成において「ありがちかどうか」を考慮することは重要となる.しかし既存研究では,Recurrent Neural Network Language Model (RNN-LM) などの高頻出の単語系列を生成する手法を用いているため,ありがちな表現が生成されてしまう.

本研究の目的は,入力した曲調にあった単語を自動生成すると同時に,ありがちでない歌詞を自動生成することである.この目的を達成するために次の 2 つの課題に取り組む.(1) 曲調に依存して単語の使用傾向が異なるかどうかを確認するために,曲調ベクトルと単語ベクトルを同一空間に埋め込んだベクトル空間モデルを教師なし学習する.(2) 曲調と単語の関係性を考慮すると同時に,ありがちでない歌詞を生成する新しいベクトル空間モデルを構築する.具体的には,曲調ベクトルと文脈単語ベクトルを入力すると,それら 2 つのベクトルと類似する単語ベクトルを生成するモデルである.この提案手法では,ありがちな単語を負例として学習するため,ありがちでない単語の生成が期待できる.

実験の結果,提案モデルは従来の RNN-LM タイプのモデルと比較して,曲調と関連した単語と,ありがちでない単語を自動生成できることが確かめられた.

2 曲調・歌詞データの準備

曲調と単語の関係性を分析・モデル化するために,本研究では 458,572 曲の英語歌詞を用意した.歌詞に対応する楽曲の曲調を音響信号から分析するために,インターネット上から視聴用の部分的な音源ファイル (30 秒固定:44.1kHz) を収集した.この対応づいたデータを「曲調・歌詞データ」と呼び,その総再生時間は約 159 日分となった.

2.1 音響信号から Bag-of-Audio-Words への変換

音響信号から曲調を抽出する手法として,Liu らが用いた Bag-of-Audio-Words (BoAW) を我々も用いた [4].BoAW とは,連続的な音響信号から離散的な擬似単語へ変換されたものである.BoAW の作成手順を以下に示す:

1. 語彙数 k の audio-word を作るために,音響信号から曲調を表す Mel-Frequency Cepstral Coefficients (MFCCs) を計算する.
2. 計算した MFCCs を 250-ms 毎に分割する.
3. 分割された各 MFCCs を *k-means* クラスタリングする.

以上の手順で得られた k 個のクラスタが audio-word に対応する.本研究では $k = 3000$ とした.

3 曲調と歌詞の関係性分析

バラードのようなゆったりとした曲調では “love” などの恋愛に関する単語が使わやすいように,曲調に依存して単語の使用傾向が変わることが予想される.しかし,このような関係性は大規模な歌詞データに対して明らかでない.そこで本研究では,「曲調に依存して単語の出現頻度に偏りが発生する」という仮定のもと,共起する audio-words と単語が近づくベクトル空間を構築することで,この関係性を分析する.

3.1 Multi-Modality Vector Space Model

本研究では,分布仮説 [5] に基づき,曲調と単語の類似性を捉えたベクトル空間を構築するために,Skip-Gram with Negative-Sampling (SGNS) [6] を拡張する.拡張モデルは曲調ベクトルと単語ベクトルを同一空間に埋め込んだモデルであり,Multi-Modality Vector Space Model (MM-VSM) と呼ぶ(図1).

表1　MM-VSM における対象の単語と類似度が大きい単語・楽曲.括弧の中に iTunes に登録されているジャンルタグを示す.

	入力単語ベクトル $\boldsymbol{v}(w)$					
	$\boldsymbol{v}$(sweet)		$\boldsymbol{v}$(kill)		$\boldsymbol{v}$(family)	
類似単語	lover	0.88	murder	0.91	young	0.91
	love	0.87	hate	0.90	grown	0.90
	lips	0.86	feed	0.89	paid	0.90
	kisses	0.86	enemies	0.89	short	0.90
	blue	0.85	crush	0.88	house	0.89
類似楽曲	Please(Traditional Pop)	0.41	Virunus(Metal)	0.48	Freestyle(Hip-Hop/Rap)	0.45
	I Wonder(R&B/Soul)	0.38	The One That Kills the Least(Metal)	0.47	Memories(Hip-Hop/Rap)	0.45
	When a Woman Loves a Man(Jazz)	0.38	Lost Symphonies(Rock)	0.47	Roll Off(Hip-Hop/Rap)	0.44
	The Lamp Is Low(Pop)	0.38	Nuclear Torment(Rock)	0.47	Them Jeans(Hip-Hop/Rap)	0.44
	The Prohibition(Vocal)	0.38	Lamnidae(Rock)	0.47	Hustlers(Hip-Hop/Rap)	0.44

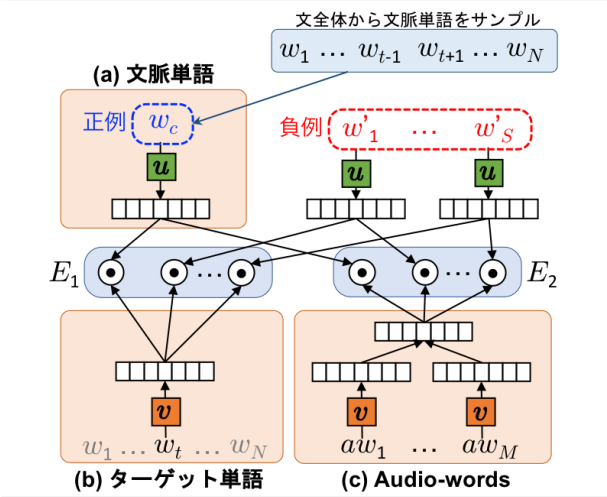

図1　Multi-Modality Vector Space Model

MM-VSM では,SGNS と同様に (a) 文脈単語 w_c のベクトルと (b) ターゲット単語 w_t のベクトルの内積が最大になるように損失関数 E_1 を学習する:

$$E_1 = -\log\sigma\Big(\boldsymbol{u}(w_c)^\mathsf{T} \cdot \boldsymbol{v}(w_t)\Big) - \sum_{s=1}^{S} \log\sigma\Big(-\boldsymbol{u}(w'_s)^\mathsf{T} \cdot \boldsymbol{v}(w_t)\Big) \quad (1)$$

ここで $\sigma(\cdot)$ はシグモイド関数であり,$\boldsymbol{u}(\cdot)$ と $\boldsymbol{v}(\cdot)$ はそれぞれ文脈ベクトルとターゲットベクトルである.S はサンプルする負例の数を示し,負例 w'_s を以下のユニグラム分布 $P(w'_s)$ からサンプルする:

$$P(w'_s) = \frac{\#(w'_s)^{0.75}}{\sum_{w' \in V}(\#(w')^{0.75})} \quad (2)$$

ここで $\#(w')$ は単語 w' の出現頻度であり,V は単語の語彙である.なお,通常の SGNS ではターゲット単語の周囲の単語を文脈単語 w_c としてサンプルするが,本研究では文全体から文脈単語をサンプルする.このように文脈単語をサンプルすることで,単語の Topical な類似性が学習されることが知られている [7].

さらに MM-VSM では E_1 の学習と同時に,(c)audio-word aw のベクトルの平均と (a) 文脈ベクトルの内積も最大となるように損失関数 E_2 を計算する:

$$E_2 = -\log\sigma\Big(\boldsymbol{u}(w_c)^\mathsf{T} \cdot \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{v}(aw_m)\Big) - \sum_{s=1}^{S} \log\sigma\Big(-\boldsymbol{u}(w'_s)^\mathsf{T} \cdot \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{v}(aw_m)\Big) \quad (3)$$

ここで M は 1 曲内の audio-word の数である.学習では 2 つの目的関数 E_1 と E_2 を交互に計算する.分析では,ベクトルの次元数を 300 とし,Negative Sampling 数は 20 とした.最適化アルゴリズムには SGD を用い,学習率は 0.001,Epoch 数は 5 とした.

3.2　分析結果

学習した MM-VSM を用いて,3 つの単語ベクトル $\boldsymbol{v}$(sweet),$\boldsymbol{v}$(kill),$\boldsymbol{v}$(family) と類似度が高い単語と楽曲のベクトル $\frac{1}{M}\sum_{m=1}^{M} \boldsymbol{v}(aw_m)$ を計算し,その結果を表1に示す.この表より,入力単語 “sweet” に対して “love” や “kisses” などの恋愛に関する単語との類似度が高いことがわかる.また単語 “sweet” は R&B や Jazz などのゆったりとした楽曲との類似度が高い.他の入力単語も同様に,単語と曲調の類似性と捉えていることがわかる[*1].このように,本研究の曲調・歌詞データ中の曲調と単語には関係性が存在し,その類似性を MM-VSM が学習できていることを定性的に確認できた.このような知見から,我々は曲調にあった歌詞を生成するモデルを構築する.

4　単語のありがちさと曲調を考慮したベクトル空間モデル

本節では,入力楽曲の曲調にあい,ありがちでない歌詞を生成するモデルを提案する.モデルの入力として,楽曲の BoAW$\{aw_1, \ldots aw_M\}$ が与えられる.そして時刻 t までの単語系列 $w_1, \ldots w_{t-1}$ を入力し,単語 w_t を生成する.

ここで技術的な課題として,ありがちでない単語のモデル化が挙げられる.従来の RNN-LM は高頻度の単語系列の生成確率を高くするため,ありがちなフレーズを生成

[*1] 表 1で示した例は`https://kentow.github.io/anlp19/`にて公開中である.

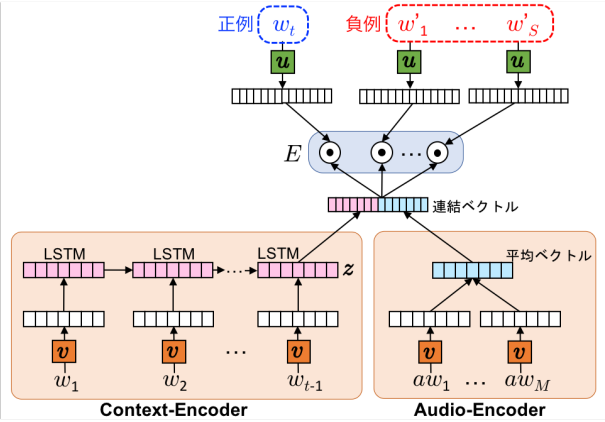

図2　Hybrid Vector Space Model

してしまう.本研究ではこの課題点を解決するために,Context2Vec [8] における Negative Sampling に着目した.Context2Vec は,RNN を用いて入力文脈 $w_1, ...w_{t-1}$ をベクトル z へ変換し[*2],その z と生成単語ベクトル $\boldsymbol{u}(w_t)$ を近距離に写像するベクトル空間モデルである.この Context2Vec の学習では,SGNS と同様に,単語ユニグラム分布から負例をサンプルするため,ありがちな単語ベクトルとの距離を遠ざける効果が期待される.

もう一つの課題として,曲調と単語の関係性のモデル化が挙げられる.本研究では,MM-VSM と同様に audio-word ベクトルの平均 $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)$ と単語ベクトル $\boldsymbol{u}(w_t)$ を近距離に写像するように Context2Vec を拡張した.

4.1　モデル構成

提案モデルは Context2Vec を基本とする.Context2Vec では,RNN を用いて文脈 $w_1, ...w_{t-1}$ をベクトル $\boldsymbol{z}$ へ変換し,生成する単語 w_t のベクトル $\boldsymbol{u}(w_t)$ との内積を最大化するように以下の損失関数を計算する:

$$E_{c2v} = -\log\sigma\Big(\boldsymbol{u}(w_t)^{\mathsf{T}}\cdot\boldsymbol{z}\Big) - \sum_{s=1}^{S}\log\sigma\Big(-\boldsymbol{u}(w'_s)^{\mathsf{T}}\cdot\boldsymbol{z}\Big) \quad (4)$$

ここで w'_s は負例を表し,式 (2) と同様のユニグラム分布 $P(w'_s)$ からサンプルされる.つまり頻出単語が負例になりやすいため,文脈ベクトル $\boldsymbol{z}$ とありがちな単語のベクトル $\boldsymbol{u}(w'_s)$ の距離が遠ざかるように学習される.

本研究では,この Context2Vec と曲調ベクトルを組み合わせることで,ありがちでなく,曲調と関連性を持つ単語を生成する Hybrid-VSM を提案する(図2).具体的には RNN の出力ベクトル $\boldsymbol{z}$ と曲調ベクトル $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)$ の両方が,単語ベクトル $\boldsymbol{w_t}$ と近づくように損失関数を学習する:

$$E = -\log\sigma\Big(\boldsymbol{u}(w_t)^{\mathsf{T}}\cdot[\boldsymbol{z}, \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)]\Big)$$
$$-\sum_{s=1}^{S}\log\sigma\Big(-\boldsymbol{u}(w'_s)^{\mathsf{T}}\cdot[\boldsymbol{z}, \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)]\Big) \quad (5)$$

[*2] 通常の Context2Vec は前後の文脈 $w_1, ...w_{t-1}, w_{t+1}, ...w_T$ を双方向 RNN でベクトル化しているが,本研究では前文脈のみを単一方向 RNN でベクトル化する.

表2　単語穴埋めタスクの結果(大きいほど性能が良い)

	RNN-LM	Enc-Dec	Context2Vec	Hibrid-VSM
モデルタイプ	RNN-LM		VSM	
曲調ベクトル	無	有	無	有
RANDOM	**24.9**	24.8	17.6	17.0
DF	**42.1**	41.8	26.2	19.0
IDF	3.0	2.9	3.6	**3.8**
MUSIC	3.4	3.5	3.7	**4.6**

ここで $\boldsymbol{z}$ は文脈単語列 $w_1, ...w_{t-1}$ を RNN で変換した最終層を表す.また $[\boldsymbol{a}, \boldsymbol{b}]$ はベクトル $\boldsymbol{a}$ と $\boldsymbol{b}$ の連結を意味する.入力ベクトル $\boldsymbol{v}(\cdot)$ 及び $\boldsymbol{z}$ の次元数を d とした時,生成単語ベクトル $\boldsymbol{u}(\cdot)$ の次元数は $2d$ となる.

5　実験

本研究では,モデルがありがちでない単語や曲調とあう単語を生成できるか評価するために,以下の 4 つの単語穴埋めタスクを新たに設計する.

RANDOM　テストセットの各文から空欄となる単語をランダムに決め,モデルに空欄を予測させる.この指標では,出現頻度が高い単語が空欄に選ばれやすいため,ありがちな単語生成の性能を評価することを意味する.

DF　各行の文書頻度が最も大きい単語を空欄とし,モデルに空欄を予測させる.この指標も,ありがちな単語の生成性能を評価することを意味する.

IDF　各行の文書頻度が最も小さい単語を空欄とし,モデルに空欄を予測させる.この指標は,ありがちでない単語の生成性能を評価することを意味する.

MUSIC　3節で提案した MM-VSM を用い,曲調ベクトル $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)$ と最も類似した単語を空欄とし,モデルに空欄を予測させる.この指標では,曲調と関係を持つ単語をモデルが生成できるかを確かめる.

なお,本タスクは選択肢から正解を選ぶタスクではなく,全語彙から正解を予測するタスクとなる.単語穴埋めタスクの精度として,以下のスコアを計算する:

$$score = \frac{\text{モデルが正解を予測した問題数}}{\text{総問題数}} \times 100 \quad (6)$$

5.1　比較手法

提案モデルの性能を評価するために,以下の 4 モデルを比較する.(1) RNN-LM: 歌詞データのみで学習した単語列を生成する標準的な RNN-LM.(2) Encoder-Decoder: 標準的な RNN-LM を Decoder とし,RNN の初期状態に楽曲の audio-word のベクトルの平均 $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)$ を入力する.(3) Context2Vec: 歌詞データのみで学習した Context2Vec.(4) Hybrid-VSM: 本研究で提案した RNN と音響信号の合成ベクトルを埋め込んだ VSM.なお,RNN-LM タイプのモデル (1) と (2) は最も生成確率が高い単語を空欄に埋め,VSM タイプのモデル (3) と (4) は最も類似度が高い単語を空欄に埋める.

表3　Hybrid-VSM を用いた歌詞生成の例.この楽曲と生成歌詞の例は`https://kentow.github.io/anlp19/`にて公開中である.

曲名	ジャンル	先頭単語:I	先頭単語:The
Amazing Grace	Pop	I'm not asking for mercy anymore.	The sun shines bright as snow.
Killing Time	Metal	I invoke the lord above thy god.	The stench of blood runs dry.
Why I Love You	R&B/Soul	I wanna be your lover lover.	The scent of jasmine brew.

5.2　データとパラメータの設定

データを訓練・開発・テスト用に 8:1:1 の比率で分割した.また,学習対象の語彙は出現頻度 20 以上の単語を使用し,その他の単語は未知語タグに置換する.

本研究では RNN として LSTM を使用した.入力ベクトルと RNN の次元数 d を 300 とし,Negative Sampling 数は $S = 20$ とした.最適化アルゴリズムは Adam を用い,学習率は 0.001,ミニバッチ数は 100 とした.Epoch 数は 10 とし,開発セットを用いた評価で最高性能のモデルを用いた.以上の設定を全比較手法において統一した.

5.3　実験結果

表2に各単語穴埋めタスクのスコアを示す.この表より,頻出単語の予測性能(RANDOM,DF)に関しては,VSM タイプのモデルより RNN-LM タイプのモデルのほうが高スコアであることがわかる.一方で,あちがちでない単語の予測性能(IDF)に関しては,VSM タイプのモデルのほうが高スコアである.これらの結果は,VSM タイプのモデルは Negative Sampling によってありがちな単語の生成が抑制されたためだと考えられる.

曲調と関連した単語の予測性能(MUSIC)に関しては,提案した Hybrid-VSM が高性能だとわかる.これは曲調ベクトルが曲調に関連する単語予測に寄与することを意味する.しかし,Encoder-Decoder モデルは曲調ベクトルを考慮しているにもかかわらず,RNN-LM と同等の性能である.これは,Decoder の RNN が「ありがちな単語系列」の生成確率が高くなるよう強く学習してしまった結果,曲調と単語の関係性を学習できなかったためだと考えられる.

以上の結果より,提案手法はありがちでなく,曲調と関連した単語の予測能力が従来の RNN-LM タイプのモデルよりも高いことがわかった.

6　曲調に基づく自動歌詞生成

本節では,提案した Hybrid-VSM が曲調にあった歌詞を生成できるかを定性的に分析する.テストセットから選んだ異なるジャンルの楽曲と,先頭の単語 "I" または "The" を入力したときに生成される歌詞を比較する.自動歌詞生成はビーム探索を用いて,文脈ベクトル $\boldsymbol{z}$ と曲調ベクトル $\frac{1}{M}\sum_{m=1}^{M}\boldsymbol{v}(aw_m)$ の連結ベクトルと類似した上位 W 個の単語 w_t を順番に生成する.本研究ではビーム幅を $W = 3$ に設定した.

表3に生成例を示す.表より,曲調によって生成する単語に違いが見られる.例えば,Amazing Grace などの幻想的な楽曲では "mercy" や "sun shines" のような厳かなフレーズが生成され,Metal の楽曲では "blood" などの負のイメージに関する単語が生成された.このように,Hybrid-VSM は曲調にあった歌詞を生成していることが確認できた.

7　おわりに

本研究では,楽曲の曲調に合わせた歌詞生成と,ありがちでない歌詞生成を両立する新しいベクトル空間モデルを提案した.まず,歌詞の出現傾向が曲調に依存するかを分析するために,曲調と単語を同一空間へ写像したベクトル空間モデルを構築した.定性的分析の結果,曲調によって単語の使用傾向が異なることが確認できた.また,単語穴埋めタスクの実験結果より,以下の結論が得られた.(1) 出現頻度が高い単語の予測性能は RNN 言語モデルが高く,(2) 出現頻度が低い単語の予測性能はベクトル空間モデルが高い.(3) 曲調に合わせた歌詞の生成性能は文脈ベクトルと曲調ベクトルを組み合わせたベクトル空間モデルが高い.今後は,提案モデルを作詞支援システムに組み込みことを目指す.

謝辞　産業技術総合研究所の坂東宜昭氏より本研究に有益な意見をいただいた.また,本研究の一部は JST ACCEL (JPMJAC1602) の支援を受けた.

参考文献

[1] Hugo R. Gonçalo Oliveira, F. Amialcar Cardoso, and Francisco C. Pereira. Tra-la-lyrics: an approach to generate text based on rhythm. In *Proc. of 4th International Joint Workshop on Computational Creativity*, pages 47–55, 2007.
[2] Kento Watanabe, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. A melody-conditioned lyrics language model. In *NAACL 2018*, pages 163–172, 2018.
[3] Peter Potash, Alexey Romanov, and Anna Rumshisky. Ghostwriter: Using an LSTM for automatic Rap lyric generation. In *EMNLP 2015*, pages 1919–1924, 2015.
[4] Yang Liu, Wan-Lei Zhao, Chong-Wah Ngo, Chang-Sheng Xu, and Han-Qing Lu. Coherent bag-of audio words model for efficient large-scale video copy detection. In *Proc. of the ACM International Conference on Image and Video Retrieval*, pages 89–96, 2010.
[5] Zellig S Harris. Distributional structure. *Word*, 10(2–3):146–162, 1954.
[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proc. of the 26th Annual Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.
[7] Mohit Bansal, Kevin Gimpel, and Karen Livescu. Tailoring continuous word representations for dependency parsing. In *ACL 2014*, pages 809–815, 2014.
[8] Oren Melamud, Jacob Goldberger, and Ido Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proc. of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, 2016.