

# 校歌のための歌詞作成支援システムの開発

重藤優太郎<sup>1,2</sup> 西田典起<sup>1</sup> Shanshan Liu<sup>1</sup> 松本裕治<sup>1</sup>

森田敏生<sup>3</sup> 中村伊知哉<sup>4</sup> 石戸奈々子<sup>4</sup>

<sup>1</sup> 理化学研究所 AIP <sup>2</sup> 千葉工業大学 STAIR Lab

<sup>3</sup> 総和技研 <sup>4</sup> iU 情報経営イノベーション専門職大学 B Lab

shigeto@stair.center {noriki.nishida,shanshan.liu,yuji.matsumoto}@riken.jp

morita@sowa.com nakamura@ichiya.org nanako@ishido.org

## 概要

本論文では、校歌のための歌詞作成支援システムを紹介する。前提として、システム単独で歌詞を生成する(完成させる)というよりは、ユーザとのやりとりを通して歌詞を完成させること(作詞の支援)を想定している。そのため、システムによる歌詞の生成とユーザの編集を交互に行えるように設計した。校歌の特徴として、(ユーザが希望した場合)学校名を歌詞に含めることが要求されるので、制約付きデコーディングによってこれを実現した。

## 1 はじめに

理化学研究所 AIP および iU 情報経営イノベーション専門職大学 B Lab が共同で取り組んでいる「超校歌～AIがつくるみんなの校歌～」プロジェクト<sup>1)</sup>では、AI技術を用いて、価値観が多様化する時代に相応しい、進化する校歌のあり方を検討している。この取り組みの一環として、校歌のための歌詞作成支援システムを開発した。本論文では、今回開発した歌詞作成支援システムの紹介を行う。

本システムは、校歌の歌詞作成支援を目的としており、特に作詞経験の少ないユーザの支援を想定している。作詞経験が少ないユーザにとって、ゼロから歌詞を作ることは難しいと思われる。一方で、原案となる歌詞が提示された場合、それを編集し、完成を目指すことは比較的容易である。本システムでは、原案を提示することによって、経験が少ないユーザであっても作詞に簡単に取り組めるようになる(と期待している)。

これを踏まえた上で、以下の4点をシステムの要件とした。

- システムが原案となる歌詞を提案できる。
- ユーザが提案された歌詞を自由に編集できる。
- システムが、ユーザの編集に基づいて、歌詞を再生成(修正)することができる。
- 生成する歌詞に関する制約条件を入力できる。

この要件を満たすためには、歌詞生成器を開発する必要がある。本システムでは、キーワード(歌詞の雰囲気表現する単語など)と学校の種類(幼稚園・保育園、小学校、中学校、高校、大学のうち、どの区分の校歌を生成するか)がユーザによって入力されることを前提としている。さらに、ユーザが希望した場合、特定の単語(やフレーズ)を必ず歌詞に含めるといった制約を付け加えることができる。本システムでは、これらの情報から歌詞を生成する。

歌詞生成と一般的な言語生成を比較した場合、歌詞生成固有の問題として、上述した特定の単語を必ず歌詞に含める制約の他にも、韻やモーラ数などに関する制約が考えられる。本論文では特定の単語(学校名<sup>2)</sup>)を歌詞に含めることのみを取り扱い、残りの問題(韻とモーラに関する制約)は今後の課題とした。以後、「学校名制約」と記述した場合、学校名を歌詞に含めるか否かに関する制約を指すことにする。

## 2 校歌歌詞作成支援システム

本システムでは、ユーザとのやりとりを通して歌詞を完成させること(作成の支援)を目指している。これを実現するため、システムによる歌詞生成とユーザの編集を交互に行えるように設計した。具体的には、以下の手続きによって歌詞を作成する。

1. ユーザがキーワード(歌詞の雰囲気表現する単語など)や学校の種類(幼稚園・保育園、小学

1) <https://blaboratory.org/chokouka/>

2) 地名などを指定することも考えられるが、本研究では学校名のみを対象とした。

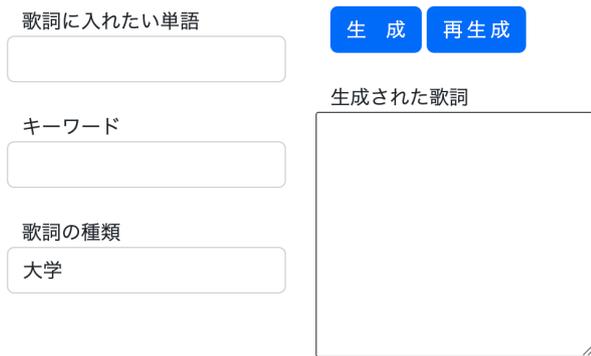


図1 ユーザインターフェース。

1. 校, 中学校, 高校, 大学のどれか), 学校名制約 (学校名を歌詞に含めるか否か) を入力する
2. システムがユーザーの入力に基づき原案となる歌詞を生成する
3. ユーザが生成された歌詞を編集する
4. システムがユーザーの編集に応じて歌詞を再生成 (修正) する
5. 3-4 を繰り返す

まずはじめに, システムのユーザインターフェースと歌詞生成器を説明し, その後, 実際にシステムを用いて作成した歌詞の例を示す。

## 2.1 ユーザインターフェース

図1に開発したシステムのインターフェースを示す。このシステムには, 2個のテキストボックスと1個のプルダウン型選択メニューが準備されている。各テキストボックスには, 歌詞に含める単語 (学校名) と歌詞の雰囲気を表すキーワードを入力することができる。プルダウンは学校の種類 (幼稚園・保育園, 小学校, 中学, 高校, 大学) を選択することができる。ユーザが, これらの情報を入力し, 生成ボタンをクリックすることで, システムが歌詞を生成する。生成された歌詞は右のテキストボックスに表示される。ユーザはこの歌詞を自由に編集することができる。編集後, 再生成ボタンをクリックすることで, 編集箇所より後ろの歌詞をシステムが再生成 (修正) する。この再生成時には, ユーザの入力 (テキストボックスとプルダウン) も反映されるため, 歌詞の途中からキーワードや学校の種類を変更することもできる。

## 2.2 歌詞生成器

本論文で取り扱う歌詞生成は, ユーザの入力 (キーワード, 学校の種類, 学校名制約) から歌詞を生成す

るタスクである。これは, 言語生成タスクの1種だと考えることができる。近年の言語生成においては, エンコーダ・デコーダモデルを採用することが主流であり, その有効性が種々の言語生成タスクにおいて報告されている [1, 2, 3]。このような背景から, 本研究においてもエンコーダ・デコーダモデルを用いて歌詞の生成を行う。具体的には, キーワードと学校の種類を特殊記号 [SEP] で連結したものをエンコーダに入力し, デコーダによって歌詞を生成する。再生成時 (ユーザが歌詞を編集した場合) には, 歌詞の先頭からユーザが編集した箇所までをデコーダの出力として固定し, 後続 (続き) の歌詞を生成する。

ユーザが歌詞に含めたい単語 (学校名) を指定した場合, デコーディングは制約付き生成問題となる。制約付き生成問題は, 制約を必ず満たす方法 [4, 5, 6, 7, 8] と必ずしも満たさない (が満たされやすくなるように工夫をする) 方法 [9] の2種類のアプローチが存在する。本システムにおいては, 学校名が指定された場合, その指定された学校名を歌詞に必ず含めることが望ましいので, 前者のアプローチを採用する。具体的には, 制約付きビームサーチ [4, 5, 6, 7] を利用する。

当然ではあるが, ある学校の歌詞を考える場合, その歌詞に他の学校の名前が含まれることは考えにくい (例えば, 千葉工大の校歌を考えた場合, 「IU 情報経営イノベーション専門職大学」が歌詞に含まれることは考えにくい)。一方で, 学校名を歌詞のどこに配置するかには, なんらかのパターンが存在していることが予想される (実際, 多くの校歌において, 歌詞の最後に学校名を配置する傾向があることを観測している)。そのため, 学校名制約が課されている場合, 歌詞生成器が学習すべきものは, 学校名 (の文字列) 自体ではなく, どこに学校名を配置するかという点だと考えられる。これを実現するため, データの前処理として, 各歌詞に出現する学校名を特殊記号 [NAME] に置換した。これによって, 学校名の表層情報を削除し, 位置情報のみ (どこに配置すればよいか) をモデルに学習させる。生成時には, 制約付きビームサーチによって [NAME] を含む系列 (歌詞) を探索する。後処理として, 生成された歌詞に含まれる [NAME] を実際の学校名 (ユーザの入力) に置換する。図2に具体例を示す。

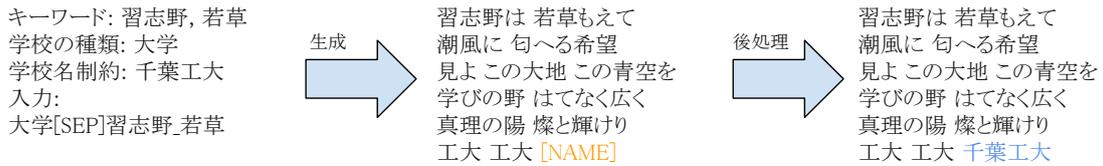


図2 歌詞生成手続きの例. 特殊記号 [NAME] (オレンジ色) を生成し, 後処理で学校名 (青色) に置換する.

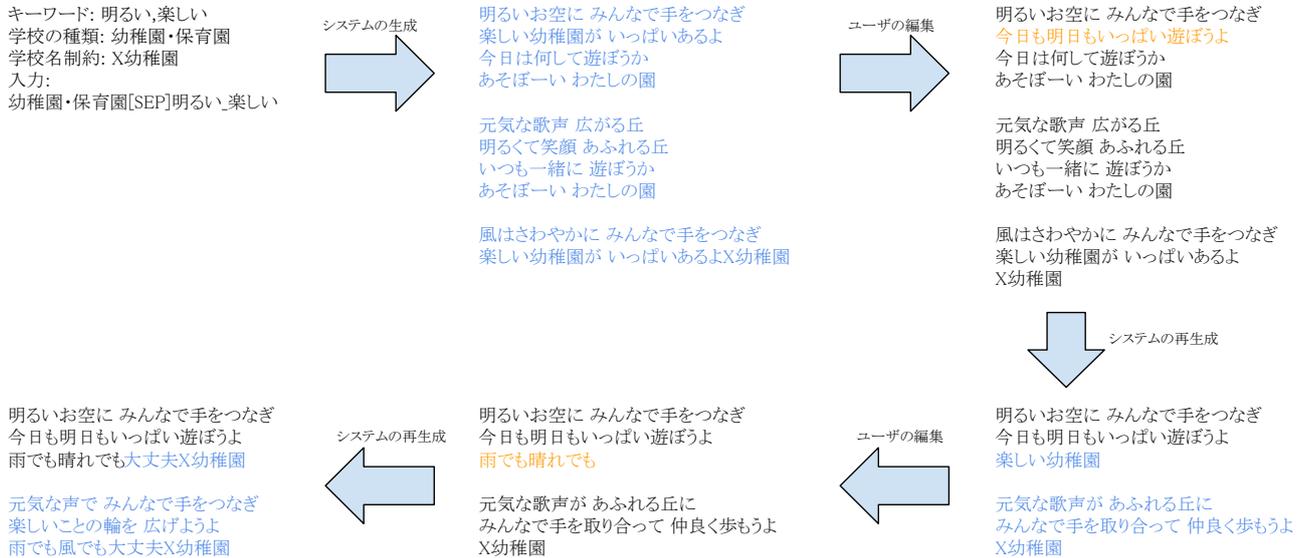


図3 システムの使用例: システムの生成 (青色) とユーザの編集 (オレンジ色) を交互に繰り返すことができる.

表1 実験で使用したデータの曲数と歌詞の文字数.

	曲数			歌詞の文字数		
	訓練	開発	評価	min	mean	max
幼稚園・保育園	37	5	5	57	148	247
小学校	140	18	18	33	145	243
中学校	143	18	18	38	148	262
高校	153	19	20	34	168	990
大学	138	17	18	28	160	319

歌詞

習志野は 若草もえて  
 潮風に 匂へる希望  
 見よこの大地 この青空を  
 学びの野 はてなく広く  
 真理の陽 燦と輝けり  
 工大 工大 千葉工大

前処理後

習志野は 若草もえて  
 潮風に 匂へる希望  
 見よこの大地 この青空を  
 学びの野 はてなく広く  
 真理の陽 燦と輝けり  
 工大 工大 [NAME]

図4 データの前処理: 学校名の置換 (オレンジ色) とキーワードの抽出 (青色).

## 2.3 デモンストレーション

図3に, 本システムの使用例を示す. これまで説明した通り, システムによる歌詞の生成とユーザによる編集を交互に繰り返すことで, 作詞に取り組むことができる. 編集および再生成によって, 後続 (オレンジ色の後) の歌詞が変化することがわかる.

## 3 歌詞生成器の性能評価

本システムで用いた歌詞生成器の性能を報告する.

### 3.1 実験設定

本実験では, 校歌 767 曲からなるデータを使用した. このデータは, 幼稚園・保育園 (47 曲), 小学校

(176 曲), 中学校 (179 曲), 高校 (192 曲), 大学 (173 曲) の校歌で構成されており, 各曲は歌詞とメタ情報 (キーワード, 学校の種類, 学校名制約) のペアになっている. このデータをランダムに分割し, 80% (611 曲) を訓練用に, 10% (77 曲) を開発用, 残り (79 曲) を評価用のデータとした. 本実験で使用したデータの統計情報を表1に示す.

実際のシステム運用時には, メタ情報はユーザによって入力されることを想定しているが, 性能評価を簡便に行うため, 以下の方法でメタ情報を準備した. 歌詞から名詞, 形容詞, 副詞を抽出し, それらをキーワードとした. 学校の種類は, 実際の種類 (幼稚園・保育園, 小学校, 中学校, 高校, 大学のうちのどれか) を使用した (例えば, 千葉工大の校歌の場合,

## 1エピソード目

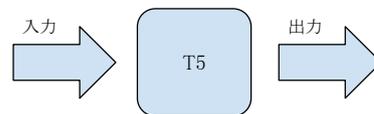
キーワード: 習志野, 若草  
学校の種類: 大学  
学校名制約: 千葉工大  
入力:  
大学[SEP]習志野\_若草



習志野は 若草もえて  
潮風に 匂へる希望  
見よこの大地 この青空を  
学びの野 はてなく広く  
真理の陽 燦と輝けり  
工大 工大 [NAME]

## 2エピソード目

キーワード: 習志野, 潮風, 希望  
学校の種類: 大学  
学校名制約: 千葉工大  
入力:  
大学[SEP]習志野\_潮風\_希望



習志野は 若草もえて  
潮風に 匂へる希望  
見よこの大地 この青空を  
学びの野 はてなく広く  
真理の陽 燦と輝けり  
工大 工大 [NAME]

図5 学習時の入力と出力. キーワードを全て使うのではなく, エピソードごとにランダムにサンプルする.

表2 実験結果. 各スコアは平均値, かつこの中は標準偏差を示している..

R-1	R-2	R-L	FP
26.9 (0.6)	3.6 (0.3)	16.7 (0.5)	37.4 (0.2)

「大学」とした). 歌詞に学校名が含まれている場合は学校名制約を課すものとし, 学校名が含まれていない場合, 制約はないものとして取り扱った. 制約のある曲は, 学習データに 189 曲, 開発データに 32 曲, 評価データに 26 曲含まれていた. なお, データの前処理として, 各歌詞に出現する学校名を特殊文字 ([NAME]) に置換した. 図4に具体例を示す.

実際の運用を考えた場合, ユーザがキーワードを大量に入力することは稀だと思われる. そのため, 準備したキーワード全てを入力とするのではなく, 1 から 3 個ランダムに抽出し, それらを入力とすることで実際の設定に近づけた. このサンプリングは, 学習と評価の双方で行なった. 図5に学習時の例を示す.

本実験の生成対象は日本語なので, mC4 [2] および Wiki-40B [10] に含まれる日本語テキストのみで事前学習された T5 [2] を使用した.<sup>3)</sup> このモデルに対し, クロスエントロピー損失を用いてファインチューニングを行なった. 使用したハイパーパラメータおよび実装の詳細は付録に記載した.

## 3.2 実験結果

表2に ROUGE [11] を示す. 表の FP (false positive) は, 制約がない ([NAME] が正解の歌詞に含まれていない) 場合に [NAME] がどのくらい出力されたかを示しており, 値が大きいほど誤って [NAME] を出力していることを意味する. 制約がない場合であっても, 頻繁に [NAME] が出力されるのではないかと予

3) <https://huggingface.co/megagonlabs/t5-base-japanese-web>

想していたが, 本実験では, 37.4% (26 曲中で約 10 曲) であった.

本実験では, ユーザの入力となるキーワードを, 歌詞から抽出した単語のランダムサンプリングで代用している. このランダムサンプリングの影響を考慮するため, 5 回評価を行い, その平均と標準偏差を報告している. 表の通り, 標準偏差は 1.0 以下になっており, サンプリングがスコアに対して与える影響は大きくないと思われる.

## 4 関連研究

歌詞生成に関する研究は近年活発に研究されている [12, 13, 14]. 一般的な歌詞を対象とするのではなく, ラップの生成 [15] や詩 [16], 短歌の生成など [17] 様々な形態の詩を対象とした研究も行われている. また, 本研究では, キーワードから歌詞の生成に取り組んだが, メロディから歌詞を生成する研究も存在する [12, 13, 14]. 本研究は, 校歌を対象としている点, および, 生成の支援を目的としている点がこれらの研究とは異なる.

## 5 おわりに

本論文では, 校歌のための歌詞作成支援システムを紹介した. 作詞経験が少ないユーザでも気軽に作詞体験を楽しめるように, システムが原案となる歌詞を提示できるようにした. 提示された歌詞は, ユーザが自由に編集でき, その編集に応じた歌詞をシステムが再生成できるように設計した.

今後の課題は, システムではなくユーザが原案を提示すること, 歌詞生成時に韻を考慮すること, モーラ数に制約を入れること, 音・メロディから歌詞を生成すること, などの機能を実装することである.

## 謝辞

本研究を進める過程で、理研 AIP 杉山将氏、浜中雅俊氏、本多右京氏、および、iU 情報経営イノベーション専門職大学 増田知香氏、林純輝氏、木内来萌氏、牧野友季氏、森井創氏、にシステムおよび歌詞データの構築に関してご助言・ご協力いただきました。この場を借りて感謝申し上げます。

## 参考文献

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems**, 2017.
- [2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, Vol. 21, No. 140, pp. 1–67, 2020.
- [3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7871–7880, 2020.
- [4] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Guided open vocabulary image captioning with constrained beam search. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 936–945, 2017.
- [5] Matt Post and David Vilar. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1314–1324, 2018.
- [6] J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. Improved lexically constrained decoding for translation and monolingual rewriting. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 839–850, 2019.
- [7] Zhongyang Li, Xiao Ding, Ting Liu, J. Edward Hu, and Benjamin Van Durme. Guided generation of cause and effect. In **Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence**, p. 3629–3636, 2021.
- [8] 帖佐克己, 森下睦, 永田昌明. 入力拡張と制約付きデコーディングによる語彙制約付き機械翻訳. 自然言語処理, Vol. 29, No. 4, pp. 1052–1081, 2022.
- [9] Guanhua Chen, Yun Chen, Yong Wang, and Victor OK Li. Lexical-constraint-aware neural machine translation via data augmentation. In **Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence**, pp. 3587–3593, 2021.
- [10] Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. Wiki-40B: Multilingual language model dataset. In **Proceedings of the 12th Language Resources and Evaluation Conference**, pp. 2440–2452, 2020.
- [11] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Proceedings of the Workshop on Text Summarization Branches Out**, pp. 74–81, 2004.
- [12] Kento Watanabe, Yuichiro Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. A melody-conditioned lyrics language model. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 163–172, 2018.
- [13] Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. AI-Lyricist: Generating music and vocabulary constrained lyrics. In **Proceedings of the 29th ACM International Conference on Multimedia**, p. 1002–1011, 2021.
- [14] Zhonghao Sheng, Kaitao Song, Xu Tan, Yi Ren, Wei Ye, Shikun Zhang, and Tao Qin. SongMASS: Automatic song writing with pre-training and alignment constraint. In **Proceedings of the AAAI Conference on Artificial Intelligence**, pp. 13798–13805, 2021.
- [15] Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin L. Zhang, Tao Qin, Wei-Qiang Zhang, and Tie-Yan Liu. DeepRapper: Neural rap generation with rhyme and rhythm modeling. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing**, pp. 69–81, 2021.
- [16] Xiaoyuan Yi, Maosong Sun, Ruoyu Li, and Wenhao Li. Automatic poetry generation with mutual reinforcement learning. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 3143–3153, 2018.
- [17] 浦川通, 新妻巧朗, 田口雄哉, 田森秀明, 岡崎直観, 乾健太郎. モーラを考慮した fine-tuning による口語短歌生成. 言語処理学会 第 28 回年次大会 発表論文集, pp. 1328–1332, 2022.
- [18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations**, 2019.
- [19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In **Advances in Neural Information Processing Systems**, 2019.
- [20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 38–45, 2020.
- [21] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, 2018.
- [22] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: A Japanese tokenizer for business. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation**, pp. 2246–2249, 2018.

## A ハイパーパラメータ

以下のハイパーパラメータを用いて T5 の学習 (ファインチューニング) を行なった。

- learning rate: 5e-05
- batch size: 8
- optimizer: AdamW [18]
- lr scheduler: linear
- max epochs: 300

また、歌詞生成時には以下のハイパーパラメータを用いた。

- max length: 256
- temperature: 0.5
- repetition penalty: 4.0
- number of beams: 2

これらのハイパーパラメータは、開発セットにおける ROUGE-L のスコアに基づいて選択した。

## B 実装の詳細

歌詞生成器の実装は PyTorch v.1.13.1 [19] および Transformers v4.25.1 [20], SentencePiece [21] を利用した。キーワード抽出のために Sudachi [22] を使用し形態素解析を行なった。

制約付きビームサーチは Transformers の generate 関数 (force\_words\_ids を指定) を利用した。<sup>4)</sup>

ROUGE の計算には SumEval を使用した。<sup>5)</sup>

## C 論文に掲載した歌詞について

本論文では、例 (図 2, 4, 5) を示すため千葉工業大学校歌 (1 番のみ) を使用した。千葉工業大学校歌は大学ホームページにて公開されている。<sup>6)</sup> なお、本システムを構築するために使用したデータ (表 1) に、千葉工業大学校歌は含まれていない。図 3 で示した例は、実際にシステムを用いて生成した歌詞である。

4) [https://huggingface.co/docs/transformers/main/en/main\\_classes/text\\_generation#transformers.GenerationMixin.generate](https://huggingface.co/docs/transformers/main/en/main_classes/text_generation#transformers.GenerationMixin.generate)

5) <https://github.com/chakki-works/sumeval>

6) <https://www.it-chiba.ac.jp/institute/disclosure/school-badge/>