

# スタイル変換のための折り返し翻訳に基づく事前訓練

梶原 智之<sup>1,a)</sup> 三浦 びわ<sup>2,b)</sup> 荒瀬 由紀<sup>3,c)</sup>

**概要:** 本研究では、スタイル変換における少資源問題に対処し、わずか 1,000 文対の平行コーパスを用いて高品質なスタイル変換を実現する。同一言語内の翻訳問題であるスタイル変換は、機械翻訳とは異なり訓練用の平行コーパスを大規模に収集することが難しい。この問題に対して、先行研究ではルールベースのデータ拡張や目的のスタイルにおける機械翻訳とスタイル変換のマルチタスク学習が提案されているが、人手や特定のコーパスに依存するため他のスタイルへの拡張が困難であった。そこで我々は、任意のスタイルに適用可能な生コーパスに基づく転移学習のフレームワークを提案する。まず事前訓練では、生コーパスを折り返し翻訳した疑似言い換えコーパスを用いて、スタイルを考慮しない言い換え生成器を訓練する。続いて再訓練では、小規模な平行コーパスを用いて、言い換え生成器を目的とするスタイルへのスタイル変換に特化させる。丁寧さおよび難しさのスタイル変換における評価実験の結果、提案手法がスタイル・ドメイン・モデル構造のいずれにも依存せず、常に大幅に性能を改善できることを確認した。さらに詳細な分析の結果、高品質な折り返し翻訳を利用できない設定やターゲットドメインの生コーパスを利用できない設定でも提案手法は有効に機能することがわかった。

## 1. はじめに

スタイル変換は、入力文の意味を保持しつつ意味以外の情報を制御する言い換え生成の一種である。スタイル変換によって、言語学習者にとって難解な表現を平易に言い換える [1,2]、情報抽出や機械翻訳の前処理として性能を改善する [3,4]、などの恩恵を受けられる。スタイルには、性別 [5,6]、感情極性 [7-9]、難しさ [10,11]、丁寧さ [12,13] など多様な側面があるが、本研究では丁寧さに焦点を当てる。

スタイル変換を含む言い換え生成は、同一言語内の機械翻訳の問題と捉えられる [14-16]。機械翻訳では一般に、100 万文対を超える大規模な平行コーパス上で系列変換モデルを訓練する。しかし、機械翻訳のために必要な対訳コーパスが日々の生活の中で大量に生産および蓄積される一方で、言い換え生成のために必要な単言語平行コーパスが自然に作られることは期待できない。そのため、スタイル変換において利用できるのは、10 万文対ほどの小規模な平行コーパス [12,17] のみである。このような小規模な平行コーパスからは十分な書き換え規則を獲得できず、既存のスタイル変換器は入力文の多くの部分をそのまま出力する消極的な編集をしてしまう [13]。

高品質なスタイル変換を実現するために、先行研究ではルールベースのデータ拡張 [12] や目的のスタイルにおける機械翻訳とスタイル変換のマルチタスク学習 [13] が提案されている。しかし、これらの手法は人手や特定の対訳コーパスに依存するため、他のスタイルへの拡張が困難である。

そこで我々は、任意のスタイルに適用可能な生コーパスに基づく転移学習を用いて、スタイル変換における少資源問題に対処する。スタイル変換における理想的な言い換えは、入出力間での意味の等価性・出力文の文法性・出力文のスタイルの妥当性の 3 つを満たす [12,13]。本研究では、スタイルの妥当性については目的とするスタイルの平行コーパスから学習する必要があるが、意味の等価性と文法性は他のコーパスからでも学習可能と仮定する。この仮定に基づき、目的とするスタイルに依存しない生コーパスを用いて言い換え生成器を事前訓練し、まず意味の等価性と文法性を学習する。その後、事前訓練された言い換え生成器を目的とするスタイルの平行コーパス上で再訓練することで、スタイルの変換規則を学習する。このような転移学習によって、提案手法では平行コーパスが 1,000 文対と非常に少ない設定においても、高品質なスタイル変換を実現できる。

本研究では、自己符号化を用いる手法と折り返し翻訳を用いる手法の 2 種類の事前訓練を提案する。自己符号化を用いる事前訓練では、任意の生コーパスから得られる入力文をそのまま出力する言い換え生成器を訓練する。大規模

<sup>1</sup> 大阪大学データビリティフロンティア機構

<sup>2</sup> 株式会社 AI Samurai

<sup>3</sup> 大阪大学大学院情報科学研究科

a) kajiwara@ids.osaka-u.ac.jp

b) miura@aisamurai.co.jp

c) arase@ist.osaka-u.ac.jp

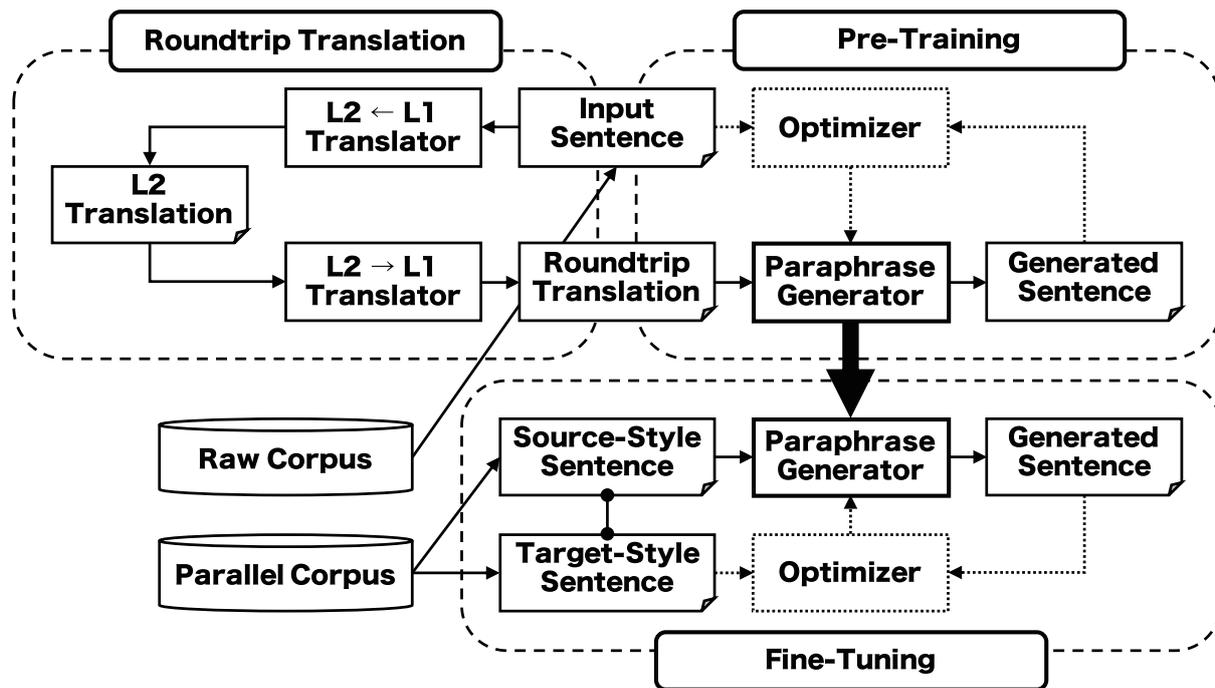


図 1 折り返し翻訳に基づく事前訓練と小規模なパラレルコーパスを用いた再訓練

な生コーパスを用いて多様な表現を学習した言い換え生成器は、パラレルコーパス上での再訓練によって高品質なスタイル変換器になる。高品質な機械翻訳器を利用できる場合、生コーパスの折り返し翻訳によって構築された疑似言い換えコーパス上での事前訓練によって、言い換え生成器が多様な同義表現を学習できる。パラレルコーパス上での再訓練では、事前訓練によって獲得された大量の書き換え規則の中から、目的とするスタイルに適した言い換えが選り出され、高品質なスタイル変換器が得られる。

丁寧さのスタイル変換のための GYAFC データセットを用いた実験の結果、我々の手法は効果的な転移学習を実現でき、高コストなデータ拡張なしに既存研究と比較して最高性能を達成した。

## 2. 提案手法

本研究では、スタイル変換における少資源問題を解決するための転移学習の手法を提案する。スタイル変換では、入出力間での意味の等価性・出力文の文法性・出力文のスタイルの妥当性の3つの観点から適切な文を生成することが求められる [12, 13]。しかし、小規模なパラレルコーパスのみからこれらの3点を同時に学習することは困難である。そこで我々は、意味と文法を学習するための事前訓練と、スタイルを学習するための再訓練の2ステップでスタイル変換器を訓練する。我々の事前訓練はスタイル非依存の生コーパスを用いるため、先行研究 [12, 13] とは異なり、任意のスタイルに適用できる。

### 2.1 事前訓練

再訓練において小規模なパラレルコーパスから効率的にスタイルを学習するために、まず言い換え生成器を意味の等価性および文法性の観点から事前訓練する。本研究では、自己符号化を用いる手法 (AE) および折り返し翻訳を用いる手法 (RT) の2種類の事前訓練を行う。AE では、生コーパスを用いて AutoEncoder としての言い換え生成を行う。RT では、生コーパスを折り返し翻訳して構築した疑似言い換えコーパスを用いて Denoising AutoEncoder としての言い換え生成を行う。生コーパスや翻訳器はスタイルに非依存のため、本手法は任意のスタイルに適用できる。

#### 2.1.1 AE：自己符号化を用いる事前訓練

入力文をそのまま出力する AutoEncoder (自己符号化) としての言い換え生成を事前訓練として実施する。自己符号化は任意の生コーパスを用いて訓練でき、意味を保持することと文法的に正しいことの両方の条件を満たす。

#### 2.1.2 RT：折り返し翻訳を用いる事前訓練

図 1 に示すように、L1 → L2 方向と L2 → L1 方向の各翻訳器を用意し、生コーパス全体を折り返し翻訳する。折り返し翻訳は、入力文に対してある種のノイズを加える操作であると考えられる。これらのノイズには、翻訳誤りだけでなく、表 1 に示すような言い換えも含まれる。そこで、折り返し翻訳から入力文を復元する Denoising AutoEncoder としての言い換え生成を事前訓練として実施する。この事前訓練によって、言い換え生成器は意味的に等価な書き換えを学習できる。なお、折り返し翻訳によって生成された文には文法性の保証がないため、翻訳文を事前訓練のソース文、元の文をターゲット文として利用する。

表 1 折り返し翻訳としての言い換えの実例

入力文	折り返し翻訳
I love watching the show.	I love to see the show.
Thanks for asking the question.	Thank you for the question.
The key to a successful relationship is good communication.	Good communication is the key to a successful relationship.

表 2 GYAFC データセットの文対数

	訓練	Informal → Formal		Formal → Informal	
		検証	評価	検証	評価
Entertainment & Music (E&M)	52,595	2,877	1,416	2,356	1,082
Family & Relationships (F&R)	51,967	2,788	1,332	2,247	1,019

## 2.2 再訓練

再訓練では主に、入力文に対してスタイルを付与することを学習する。図 1 に示すように、事前訓練された言い換え生成器を目的とするスタイルの平行コーパスを用いて通常の教師あり学習の要領で再訓練し、スタイル変換器を構築する。ここで、スタイル変換器は事前訓練された言い換え生成器から全てのパラメータを受け継ぐ。

折り返し翻訳を用いる事前訓練では、翻訳誤りを含む疑似言い換えコーパスを用いて言い換え生成器が訓練されている。再訓練では目的とするスタイルの平行コーパスを用いるため、翻訳誤りや目的とするスタイルに適さない言い換への尤度が下がり、目的とするスタイルに適した言い換えを行う高品質なスタイル変換器が得られる。

## 3. 実験設定

表 2 に示す GYAFC データセット [12] を用いて、丁寧さのスタイル変換における提案手法の有効性を検証する。

### 3.1 スタイル変換に関する実験設定

スタイル変換の実験のために、Moses ツールキット\*1 [18] によって normalize および tokenize した GYAFC データセットを用いた。GYAFC データセットは、フォーマルな英文 (Formal) とカジュアルな英文 (Informal) からなる単言語平行コーパスである。これらの文は、Yahoo Answers L6 コーパス\*2のうち、Entertainment & Music (E&M) および Family & Relationships (F&R) のドメインから抽出されたものである。なお、質問文や URL、5 単語以下の短文や 25 単語以上の長文は前処理として削除されている。検証セットと評価セットはマルチリファレンスになっており、ソーススタイルの文 1 文につき、対応するターゲットスタイルの文が 4 文ずつ付与されている。

事前訓練のための生コーパスも、同じく Yahoo Answers L6 コーパスから構築した。我々は GYAFC と同様の前処理を行った後、ドメインごとに 300 万文の訓練データと

3,000 文の検証データを抽出した。事前訓練および再訓練の両方で、Byte Pair Encoding\*3 [19] を用いてトークン数をドメインごとに 32,000 に制限した。

スタイル変換器には、Sockeye ツールキット\*4 [20] を用いて、Recurrent Neural Network (RNN)・Convolutional Neural Network (CNN)・Self-Attention Network (SAN) の各モデルを構築した。RNN モデルは、符号化器と復号器の両方で 1,024 次元の 4 層 LSTM を利用した。CNN モデルは、符号化器と復号器の両方で 512 次元の 8 層モデルを利用し、カーネルサイズは 3 とした。SAN モデルは、符号化器と復号器の両方で 512 次元の 6 層モデルを利用し、注意ヘッドは 8 とした。埋め込み層は全てのモデルで 512 次元とし、符号化器と復号器で重みを共有した。正則化には、埋め込み層および隠れ層において dropout を適用し、さらに layer-normalization および label-smoothing を使用した。最適化には adam を利用し、バッチサイズを 4,096 単語として 200 更新ごとに検証データで perplexity を評価し、32 回改善が見られなくなったところで学習を終了した。その他のハイパーパラメータの設定は、Sockeye の arxiv\_1217 ブランチ\*5に従った。

### 3.2 折り返し翻訳に関する実験設定

事前訓練における折り返し翻訳のために、スタイル変換器と同じ設定で SAN モデルを構築した。中間言語には大規模な対訳コーパスを利用可能なドイツ語を選び、英独翻訳器および独英翻訳器を用意した。翻訳器の学習には WMT-2017 英独タスク [21] のデータセットを用いた。訓練データには、News Commentary・Europarl・Common Crawl の約 450 万文対を利用した。検証データには newstest-2016 の 2,999 文対、評価データには newstest-2017 の 3,004 文対を使った。各翻訳器の BLEU を評価したところ、英 → 独方向は 27.6、独 → 英方向は 33.8 と、いずれも WMT-2017 優勝チーム [22] のシングルモデルを上回る性能であった。

\*1 <https://github.com/moses-smt/mosesdecoder>

\*2 <https://webscope.sandbox.yahoo.com>

\*3 <https://github.com/rsennrich/subword-nmt>

\*4 <https://github.com/aws-labs/sockeye>

\*5 [https://github.com/aws-labs/sockeye/tree/arxiv\\_1217/arxiv](https://github.com/aws-labs/sockeye/tree/arxiv_1217/arxiv)

表 3 GYAFC データセットにおける実験結果 (括弧外は BLEU<sub>OR</sub> ↑、括弧内は BLEU<sub>IO</sub> ↓)

	Informal → Formal		Formal → Informal	
	E&M	F&R	E&M	F&R
Source	49.09 (100.00)	51.03 (100.00)	29.85 (100.00)	29.85 (100.00)
Reference-0	100.00 (27.88)	100.00 (29.77)	100.00 (15.05)	100.00 (15.64)
Reference-1	100.00 (26.46)	100.00 (26.69)	100.00 (11.03)	100.00 (10.45)
Reference-2	100.00 (26.58)	100.00 (28.53)	100.00 (11.10)	100.00 ( 9.91)
Reference-3	100.00 (27.41)	100.00 (28.50)	100.00 (11.25)	100.00 (10.41)
R&T-PBMT	68.22 (51.62)	72.94 (51.56)	33.54 (61.53)	32.64 (74.01)
R&T-NMT	68.41 (54.16)	74.22 (54.66)	33.56 (52.95)	35.03 (59.57)
BiFT-Single	69.20 ( n/a )	73.52 ( n/a )	35.44 ( n/a )	37.72 ( n/a )
Ours (RNN-RT)	<b>71.14 (49.07)</b>	<b>75.73 (50.82)</b>	<b>38.51 (47.85)</b>	<b>39.79 (51.73)</b>
BiFT-Ensemble	71.36 (55.86)	74.49 (59.48)	36.18 (61.21)	38.34 (63.60)
MultiTask	72.13 (54.55)	75.37 (58.11)	38.04 (55.47)	39.09 (58.02)
Ours (RNN-RT)	<b>72.41 (48.62)</b>	<b>76.40 (51.28)</b>	<b>39.22 (48.42)</b>	<b>39.31 (52.68)</b>

### 3.3 比較手法

提案手法の性能を丁寧さのスタイル変換における先行研究と比較した。R&T-PBMT [12] は、ルールベースの手法によってデータ拡張された GYAFC データセット上で訓練されたフレーズベースの統計的機械翻訳モデル [18] である。R&T-NMT [12] は、同じくデータ拡張された GYAFC データセット上で訓練されたニューラル機械翻訳モデル [23] である。BiFT-Single および BiFT-Ensemble [13] は、多言語翻訳 [24] の要領で Informal → Formal と Formal → Informal の両方向の言い換えを単一モデルで訓練したものである。MultiTask [13] は、目的とするスタイルの機械翻訳とスタイル変換のマルチタスク学習に基づくモデルである。なお、BiFT-Ensemble および MultiTask は、シードの異なる 4 つのモデルを組み合わせたアンサンブルモデルである。

## 4. 実験結果

先行研究 [12,13] に倣って、各モデルの性能を参照文との BLEU (BLEU<sub>OR</sub>) [25] によって自動評価した。また、既存のスタイル変換器は消極的な編集しかできない傾向にあるという問題が知られている [13]。そこで本研究では、入力文からの積極的な書き換えを評価するために、入出力間の BLEU (BLEU<sub>IO</sub>) も調査した。BLEU<sub>IO</sub> が低いほど、スタイル変換器が入力文を積極的に編集していることを意味する。そのため、理想的なスタイル変換器は、高い BLEU<sub>OR</sub> と低い BLEU<sub>IO</sub> の両方を達成する必要がある。

### 4.1 先行研究との比較

表 3 に、提案手法および比較手法の評価結果を示す。ただし、括弧外のスコアは BLEU<sub>OR</sub>、括弧内のスコアは BLEU<sub>IO</sub> である。ここでは、提案手法のうち最も高い性能を達成した折り返し翻訳に基づく RNN モデル (RNN-RT)

のみを示す。上段には、スコアを解釈するための基準として、モデル出力の代わりに入力文または参照文を用いた場合の性能を示す。Source の BLEU<sub>OR</sub> は BLEU<sub>OR</sub> の下界、Reference の BLEU<sub>OR</sub> は BLEU<sub>OR</sub> の上界を意味する。中段では、先行研究のシングルモデルと我々の提案手法を比較する。ただし、BiFT-Single はシステム出力が公開されていないため、BLEU<sub>OR</sub> のみを彼らの論文 [13] から引用する。下段では、先行研究のアンサンブルモデルと我々の提案手法を比較する。ここでは、先行研究 [13] と条件を揃えるために、我々の提案手法もシードの異なる 4 つのモデルを組み合わせたアンサンブルモデルを用いる。

表 3 の結果から、スタイルやドメインによらず一貫して、提案手法が最高性能を達成することがわかる。BLEU<sub>OR</sub> の結果は、提案手法が常に最も参照文に近い言い換えを生成できることを示す。また、BLEU<sub>IO</sub> の結果は、提案手法が積極的な編集を可能にすることを示す。これらの実験結果から、折り返し翻訳に基づく事前訓練がスタイル変換器の転移学習に有効であることが確認できた。

表 4 に、E&M ドメインにおける各スタイル変換器の出力例を示す。R&T-PBMT および R&T-NMT では、Informal → Formal タスクにおいて、カジュアルな表現が出力文に残っている。これは、ルールベースのデータ拡張では全てのカジュアルな表現を正規化しきれなかったからだと考えられる。表 3 の BLEU<sub>IO</sub> の結果にも示されているように、MultiTask は消極的な編集を行う傾向にある。この傾向が表 4 の例からも確認でき、カジュアルな表現が出力文に残っている。Formal → Informal タスクでは、R&T-PBMT は非文を生成している。R&T-NMT や MultiTask の出力文は流暢ではあるが、入力文の意味を保持できていない。しかし、我々のモデルは意味的にも文法的にも正しい文を生成でき、スタイル変換に成功した。

表 4 E&M ドメインにおけるスタイル変換の例

Informal → Formal	
Source	I LOOOOOVVVVVVVEEEE this song SOOO Much!!!!!!
Reference-0	I love the song so much.
Reference-1	I very much enjoy this song.
Reference-2	I love this song very much.
Reference-3	I really love this song.
R&T-PBMT	I loovvvvvvveee this song very Much.
R&T-NMT	I loovvvvvvveee this song so Much.
MultiTask	I really enjoy VVVVVVVVEEEE this song.
Ours (RNN-RT)	I love this song very much.
Ours (SAN-RT)	I love that song so much.
Formal → Informal	
Source	I thoroughly enjoy the hair bands of the 1980s.
Reference-0	I love them old hair bands of the 80's!
Reference-1	the old hair bands of the 80's were amazing, i STILL love em
Reference-2	Those ol' 80's hair bands are my favourite.
Reference-3	80s hair bands are the best!
R&T-PBMT	I love the hair bands are THOROUGHLY + the 1980S.
R&T-NMT	I just like the hair of the brids.
MultiTask	I love the 80's hair.
Ours (RNN-RT)	I love the hair bands of the 80's.
Ours (SAN-RT)	I love the hair bands of the 80's.

表 5 GYAFC データセットにおける各提案モデルの BLEU

	Informal → Formal				Formal → Informal			
	E&M		F&R		E&M		F&R	
	事前訓練	再訓練	事前訓練	再訓練	事前訓練	再訓練	事前訓練	再訓練
RNN-Base		57.76		66.45		27.68		33.86
RNN-AE	48.55	64.58 (+ 6.82)	50.51	67.38 (+0.93)	29.26	32.19 (+ 4.51)	29.36	32.99 (-0.87)
RNN-RT	46.84	<b>69.03</b> (+11.27)	47.96	<b>74.28</b> (+7.83)	32.03	<b>38.36</b> (+10.68)	31.86	<b>39.00</b> (+5.14)
CNN-Base		48.06		60.58		24.85		30.93
CNN-AE	48.91	57.18 (+ 9.12)	50.64	51.41 (-9.17)	29.30	28.12 (+ 3.27)	29.49	31.67 (+0.74)
CNN-RT	46.96	<b>64.99</b> (+16.93)	47.94	<b>69.36</b> (+8.78)	31.33	<b>34.17</b> (+ 9.32)	31.10	<b>36.12</b> (+5.19)
SAN-Base		60.57		67.52		30.09		34.64
SAN-AE	48.54	65.57 (+ 5.00)	50.55	70.34 (+2.82)	29.25	33.45 (+ 3.36)	29.36	34.95 (+0.31)
SAN-RT	46.82	<b>69.58</b> (+ 9.01)	47.97	<b>74.19</b> (+6.67)	31.50	<b>38.84</b> (+ 8.75)	31.89	<b>38.56</b> (+3.92)

## 4.2 転移学習の効果

事前訓練手法およびモデル構造ごとの BLEU<sub>OR</sub> による評価結果を表 5 に示す。3 種類のモデルと 4 種類のタスクの全てにおいて、パラレルコーパスのみを用いて訓練するベースライン (-Base) と比較して、折り返し翻訳を用いる事前訓練 (-RT) に基づく転移学習によってスタイル変換の性能が常に大幅に改善された。具体的には、Informal → Formal タスクでは 6.67 から 16.93 ポイント、Formal → Informal タスクでは 3.92 から 10.68 ポイント改善された。

自己符号化を用いる事前訓練 (-AE) に基づく転移学習も、多くの場合 (3 種類のモデルと 4 種類のタスクの合計

12 実験のうち 10) にスタイル変換の性能を改善した。しかし、折り返し翻訳を用いる事前訓練と比較すると、その効果は限定的であった。これは、折り返し翻訳を用いる事前訓練が自己符号化を用いる事前訓練よりも多様な同義表現を学習できるからである。これらの実験結果から、折り返し翻訳を通して事前訓練用コーパスにノイズを加えることによって、より効果的に転移学習できることがわかる。

## 5. 分析と考察

本節では、高速に計算できる SAN モデルを用いて、より詳細な分析を行う。

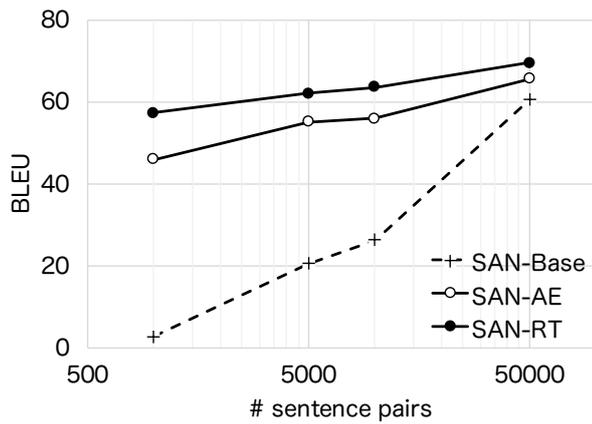


図 2 E&M タスクにおける Informal → Formal 方向の学習曲線

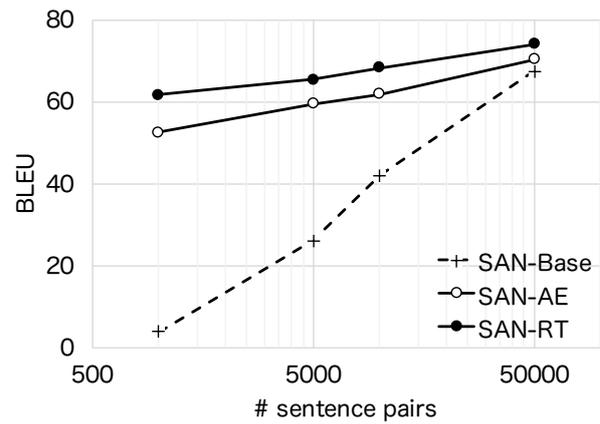


図 3 F&R タスクにおける Informal → Formal 方向の学習曲線

### 5.1 少資源設定

再訓練のために利用できるパラレルコーパスがより少ない状況での提案手法の有効性を検証する。図 2 および図 3 に、Informal → Formal 方向のスタイル変換において、再訓練用のパラレルコーパスを 1 万文対、5 千文対、1 千文対と減少させたときの性能の変化を示す。我々の転移学習によって、1 千文対のような非常に小規模なパラレルコーパスしか利用できない状況でも、高品質なスタイル変換を実現できることがわかる。特に、折り返し翻訳に基づく事前訓練を行う場合、5 千文対の再訓練によって 5 万文対の通常の教師あり学習と同等の性能を達成できた。

### 5.2 逆翻訳 vs. 折り返し翻訳

折り返し翻訳と関連する技術として、Wieting and Gimpel [26] は逆翻訳を用いて疑似言い換えコーパスを構築している。彼らは、英語とチェコ語の対訳コーパスのチェコ語側を逆翻訳し、英語の疑似言い換えコーパスを構築した。

本節では、スタイル変換における逆翻訳と折り返し翻訳の性能を比較する。我々は 3.2 節の英独対訳コーパスのドイツ語側を逆翻訳し、事前訓練のために利用した。逆翻訳を用いる事前訓練 (-BT) のために利用する文数は、AE や RT と同じく 300 万文である。

表 6 に、Informal → Formal 方向のスタイル変換における実験結果を示す。逆翻訳を用いる事前訓練は、自己符号化を用いる事前訓練よりも高い効果を発揮することがわかる。このことから、逆翻訳が折り返し翻訳に関わらず、Denoising AutoEncoder としての言い換え生成に基づく事前訓練によって効果的な転移学習が実現できると言える。これは、再訓練によって目的とするスタイルに適した言い換えが蒸留されるため、事前訓練においてできるだけ多様な同義表現を学習しておくことが重要だからである。

逆翻訳を用いる事前訓練と折り返し翻訳を用いる事前訓練を比較すると、折り返し翻訳がより高い性能を示した。

これは、折り返し翻訳の場合には任意の生コーパスを用いて事前訓練できるため、再訓練のパラレルコーパスと同じ in-domain のデータを利用できるからである。逆翻訳のためには対訳コーパスが必要であるが、任意のドメインにおいて大規模な対訳コーパスを利用できるわけではないため、逆翻訳では多くの状況で out-of-domain のデータを用いる事前訓練を行うことになってしまう。ただし、in-domain の生コーパスが大規模に利用できない状況では、折り返し翻訳を逆翻訳で代用できる。

### 5.3 その他のスタイル変換における性能：テキスト平易化

本節では、丁寧さ以外のスタイルにおける提案手法の有効性を検証する。難解な英文を平易に書き換えるテキスト平易化は、GYAFC と同程度の規模のパラレルコーパスを利用できるスタイル変換タスクである。本研究では、事前訓練のために Simple English Wikipedia<sup>\*6</sup> の 120 万文、再訓練のために WikiSmall<sup>\*7</sup> [17] の 88,837 文対および WikiLarge<sup>\*7</sup> [17] の 296,402 文対を利用した。評価には、マルチリファレンス<sup>\*8</sup> [27] を利用した。自動評価尺度には、意味の等価性および文法性だけでなく出力文の平易性も評価できる SARI [27] を用いた。

実験結果を表 7 に示す。提案手法によって、WikiSmall と WikiLarge の両方のデータセットにおいてテキスト平易化の性能が改善した。この追加の実験結果は、我々の転移学習が丁寧さ以外のスタイルにおけるスタイル変換においても有効に機能することを示す。

## 6. 関連研究

小規模なパラレルコーパスのみから高品質なスタイル変換器を訓練することは難しいため、少資源問題に対処するための手法が提案されている。Rao and Tetreault [12] は、

<sup>\*6</sup> <https://dumps.wikimedia.org/simplewiki/20181201>

<sup>\*7</sup> <https://github.com/louis martin/dress-data>

<sup>\*8</sup> <https://github.com/cocoxu/simplification>

表 6 Informal → Formal タスクにおける逆翻訳の効果

	E&M	F&R
SAN-Base	60.57	67.52
SAN-AE	65.57	70.34
SAN-BT	69.06	73.39
SAN-RT	<b>69.58</b>	<b>74.19</b>

ルールベースの手法を用いて訓練データのデータ拡張を行い、コピー機構に基づくニューラル系列変換モデル [23] を訓練した。Niu ら [13] は、目的とするスタイルの機械翻訳（フランス語からフォーマルな英語への翻訳およびフランス語からカジュアルな英語への翻訳）とスタイル変換のマルチタスク学習によって、スタイル変換の性能を更に改善した。しかし、前者はスタイルごとに人手で書き換え規則を整備する必要がありコストが高く、後者はスタイルのラベルが付与された大規模な対訳コーパスを利用できる状況は限定的である。一方で、我々の転移学習は人手のコストやスタイルのラベルに依存しない生コーパスを利用する。折り返し翻訳のための翻訳器を訓練するには対訳コーパスが必要であるが、スタイルやドメインに依存しない対訳コーパスは主要な言語では大規模に利用できる。

本研究で対象としたスタイル変換では、一般的に以下のような環境を想定する。

- (1) 目的とするスタイルにおいて、最大数 10 万文対の小規模なパラレルコーパスを利用できる。
- (2) 100 万文対を超える大規模なパラレルコーパスは、他のスタイルにおいても利用できない。
- (3) 生コーパスや対訳コーパスは豊富に利用できる。

機械翻訳におけるドメイン適応 [28] や対話応答生成における転移学習 [29] は本研究と関連するが、これらの研究では out-of-domain の大規模なパラレルコーパス（本研究では他のスタイルのパラレルコーパスに対応する）が利用できることを想定している。しかし、上述の 2 番目の設定のために、スタイル変換においてはこれらの手法を利用できない。そこで本研究では、大規模な単言語パラレルコーパスに依存しない事前訓練の手法を提案した。

一方で、パラレルコーパスを用いずにスタイル変換器を訓練する教師なしスタイル変換の関連研究 [6-9] がある。しかし、これらの研究では感情極性の変換を目的としており、必然的に入力文の意味も変化する。これらのタスクは入力文の意味を保持する言い換え生成としてのスタイル変換を目的とする本研究とは異なる。また、小規模なパラレルコーパスしか利用できないとは言え、これらの教師なし学習に基づく手法よりも教師あり学習に基づく手法の方がはるかに高い性能を達成できることが報告されている [30]。

感情極性のスタイル変換において、我々と同じく機械翻訳を利用する先行研究 [6] がある。彼らは機械翻訳の符号化器を用いて入力文の分散表現を獲得し、感情極性の分類

表 7 テキスト平易化タスクにおける SARI による評価結果

	WikiSmall	WikiLarge
SAN-Base	34.19	34.58
SAN-RT	<b>35.46</b>	<b>35.99</b>

器を用いて復号器を敵対的に訓練する。彼らの研究では、分類器を訓練するために目的とするスタイルにおける大規模なコーパスを必要とする。また、敵対的な訓練のために復号器と分類器の損失のバランスをとるハイパーパラメータの調整も必要である。我々の転移学習では、スタイルに依存しない生コーパスから事前訓練ができ、またハイパーパラメータの調整も比較的容易と言える。

## 7. おわりに

スタイル変換における少資源問題に対処するために、本研究では 2 種類の事前訓練に基づく転移学習を提案した。自己符号化に基づく事前訓練は、単純ながら多くの場合にスタイル変換の性能を改善する。折り返し翻訳に基づく事前訓練は、より効果的な転移学習を可能にし、丁寧さのスタイル変換における最高性能を更新した。後者がより有効であるのは、折り返し翻訳によって生コーパスにノイズを加えると、Denoising AutoEncoder としての事前訓練によってモデルが多様な同義表現を学習できるからである。我々の提案手法は、人手のアノテーションや特殊なデータセットに頼らないため、高い性能を発揮することに加えて低コストかつスタイル非依存な手法であるという利点も持つ。

詳細な分析の結果、1,000 文対のパラレルコーパスしか利用できない状況においても、提案手法によって高品質なスタイル変換を実現できることが明らかになった。これまで、大規模なパラレルコーパスの不足によって、マイナーな言語・ドメイン・スタイルにおける言い換え生成は困難であった。本研究では、小規模なパラレルコーパスと生コーパスから、目的とするスタイルの高品質な言い換え生成が可能であることを示した。

## 謝辞

本研究は JST (ACT-I、課題番号: JPMJPR18UB) の支援を受けたものである。

## 参考文献

- [1] Petersen, S. E. and Ostendorf, M.: Text Simplification for Language Learners: A Corpus Analysis, *Proceedings of the Speech and Language Technology in Education Workshop*, pp. 69-72 (2007).
- [2] Belder, J. D. and Moens, M.-F.: Text Simplification for Children, *Proceedings of the SIGIR Workshop on Accessible Search Systems*, pp. 19-26 (2010).

- [3] Evans, R. J.: Comparing Methods for the Syntactic Simplification of Sentences in Information Extraction, *Literary and Linguistic Computing*, Vol. 26, No. 4, pp. 371–388 (2011).
- [4] Štajner, S. and Popovic, M.: Can Text Simplification Help Machine Translation?, *Baltic Journal of Modern Computing*, Vol. 4, No. 2, pp. 230–242 (2016).
- [5] Michel, P. and Neubig, G.: Extreme Adaptation for Personalized Neural Machine Translation, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 312–318 (2018).
- [6] Prabhunoye, S., Tsvetkov, Y., Salakhutdinov, R. and Black, A. W.: Style Transfer Through Back-Translation, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 866–876 (2018).
- [7] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R. and Xing, E. P.: Toward Controlled Generation of Text, *Proceedings of the 34th International Conference on Machine Learning*, pp. 1587–1596 (2017).
- [8] Shen, T., Lei, T., Barzilay, R. and Jaakkola, T.: Style Transfer from Non-Parallel Text by Cross-Alignment, *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 6830–6841 (2017).
- [9] Fu, Z., Tan, X., Peng, N., Zhao, D. and Yan, R.: Style Transfer in Text: Exploration and Evaluation, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 663–670 (2018).
- [10] Coster, W. and Kauchak, D.: Simple English Wikipedia: A New Text Simplification Task, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 665–669 (2011).
- [11] Xu, W., Callison-Burch, C. and Napoles, C.: Problems in Current Text Simplification Research: New Data Can Help, *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 283–297 (2015).
- [12] Rao, S. and Tetreault, J.: Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 129–140 (2018).
- [13] Niu, X., Rao, S. and Carpuat, M.: Multi-Task Neural Models for Translating Between Styles Within and Across Languages, *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1008–1021 (2018).
- [14] Specia, L.: Translating from Complex to Simplified Sentences, *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language*, pp. 30–39 (2010).
- [15] Xu, W., Ritter, A., Dolan, B., Grishman, R. and Cherry, C.: Paraphrasing for Style, *Proceedings of the 24th International Conference on Computational Linguistics*, pp. 2899–2914 (2012).
- [16] Nisioi, S., Štajner, S., Ponzetto, S. P. and Dinu, L. P.: Exploring Neural Text Simplification Models, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 85–91 (2017).
- [17] Zhang, X. and Lapata, M.: Sentence Simplification with Deep Reinforcement Learning, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 595–605 (2017).
- [18] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 177–180 (2007).
- [19] Sennrich, R., Haddow, B. and Birch, A.: Neural Machine Translation of Rare Words with Subword Units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725 (2016).
- [20] Hieber, F., Domhan, T., Denkowski, M., Vilar, D., Sokolov, A., Clifton, A. and Post, M.: Sockeye: A Toolkit for Neural Machine Translation, *arXiv:1712.05690* (2017).
- [21] Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., Monz, C., Negri, M., Post, M., Rubino, R., Specia, L. and Turchi, M.: Findings of the 2017 Conference on Machine Translation, *Proceedings of the Second Conference on Machine Translation*, pp. 169–214 (2017).
- [22] Sennrich, R., Birch, A., Currey, A., Gehrmann, U., Haddow, B., Heafield, K., Miceli Barone, A. V. and Williams, P.: The University of Edinburgh’s Neural MT Systems for WMT17, *Proceedings of the Second Conference on Machine Translation*, pp. 389–399 (2017).
- [23] Jhamtani, H., Gangal, V., Hovy, E. and Nyberg, E.: Shakespearizing Modern Language Using Copy-Enriched Sequence to Sequence Models, *Proceedings of the Workshop on Stylistic Variation*, pp. 10–19 (2017).
- [24] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., Hughes, M. and Dean, J.: Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 339–351 (2017).
- [25] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation, *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (2002).
- [26] Wieting, J. and Gimpel, K.: ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 451–462 (2018).
- [27] Xu, W., Napoles, C., Pavlick, E., Chen, Q. and Callison-Burch, C.: Optimizing Statistical Machine Translation for Text Simplification, *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 401–415 (2016).
- [28] Chu, C. and Wang, R.: A Survey of Domain Adaptation for Neural Machine Translation, *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1304–1319 (2018).
- [29] Akama, R., Inada, K., Inoue, N., Kobayashi, S. and Inui, K.: Generating Stylistically Consistent Dialog Responses with Transfer Learning, *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pp. 408–412 (2017).
- [30] Luo, F., Li, P., Zhou, J., Yang, P., Chang, B., Sui, Z. and Sun, X.: A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer, *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (2019).