人工知能学会 インタラクティブ
情報アクセスと可視化マイニング研究会(第32回)
SIG-AM-32-01

発話文口調変換タスクにおける 教師なしテキストスタイル変換手法の比較検討

Comparison Study of Performance in Dialect Translation Task Using Unsupervised Text Style Transfer Methods

大原広嗣[1*] 森辰則[1]
Hirotsugu Ohara[1] Tatsunori Mori[1]

[1] 横浜国立大学大学院環境情報学府
[1] Graduate School of Environment and Information Sciences, Yokohama National University

Abstract: Appropriate control of the response dialect of a chatbot system can help the system achieve more natural and friendly conversations. We focused on unsupervised text style transfer and examined several methods, including a supervised method, for generating dialect translation models from text data with a specific dialect by deep learning, comparing the kind and amount of data used. Experimental results showed that the method of generating pseudo parallel data using back-translation showed high performance comparable to the supervised method, and Few-shot learning using ChatGPT is the most effective when the amount of data is limited. In addition, we showed that the Weighted Decoding approach proposed for conditional sentence generation can be applied to style conversion by combining it with a Denoising Autoencoder.

1 はじめに

対話システムは様々な話者の発話を含む大規模データで学習されることが多く, 生成口調の一貫性に欠ける, または「です・ます」のような単調な口調に限定されるといった問題が生じることがある. 自然な対話や親しみやすさを促進するため, 対話システムの応答を意図した口調に制御する手法が重要となっている.

口調や個性など特定のスタイルで対話システムの応答を制御することを目的とした研究として, スタイル固有の対話データを用いてシステムを転移学習する手法 [1] [2] や, システム生成文のスタイルを変換するためのスタイル変換モデルを学習する手法 [3] [4] などが提案されている. しかし, 前者の手法ではシステムの生成内容がスタイル固有のデータに影響される課題や, システムに使用される言語モデルが大規模化しており学習コストが高いといった課題が存在する. そのため本研究では後者の手法に焦点を当てた.

後者の手法はテキストスタイル変換と呼ばれ, テキストの意味内容を保ちつつ特定のスタイルに書き換えることを目的としている. スタイル変換において変換前後の対訳データが利用可能な場合, 教師ありの学習手法で変換モデルを生成することができるが, 実課題において対訳データが利用できる場面は限られるため, 目的のスタイルを持ったスタイル文のみを用いて変換モデルを学習する教師なしのスタイル変換手法が研究されている. 教師なしテキストスタイル変換は, 感情変換や形式変換といった分野で深層学習手法が盛んに研究されているが, 日本語特有の課題である口調を扱った先行研究は少ない. また, 口調を扱った先行研究の課題として, 多くの手法で (i) 独自の非公開データセットを用いており, (ii) 主観的な評価を行っているため, 手法ごとの相対的な比較が難しい現状がある.

以上を踏まえ, 本研究では複数の教師なしスタイル変換手法について, 利用できるデータ量や外部資源の利用などの条件を比較して, 性能を定量的かつ相対的に検討する. そのために, 一般に再現可能な対訳の口調スタイルデータセットを作成し, 教師あり手法をベースラインとして各手法の口調変換に対する性能の調査を行う.

*連絡先：横浜国立大学大学院環境情報学府
〒 240-8501 神奈川県横浜市保土ケ谷区常盤台 79-1
E-mail: ridivarg@outlook.com

2 関連研究

2.1 教師なしテキストスタイル変換

多くの先行研究で,GAN や VAE などをベースとしたモデルを構築し, 文の内容情報とスタイル情報を分離する手法が取られており, 先行研究 [3] [4] も分離手法を用いている. しかし, これらの手法は固定サイズ

の潜在空間を前提とするため潜在表現が可変長である Transformer モデルと相性が悪く, さらに複数種類のスタイルデータの利用を前提としている. これらの手法の中でも,Prabhumoye ら [5] は “逆翻訳には文の意味を保ちつつスタイルを弱める働きがある” という特性に着目し, スタイル文を中間言語に翻訳後, 元言語に逆翻訳した際の潜在表現を内容情報とみなし, 内容情報から元のスタイル文を復元するデコーダを敵対的に学習する手法を提案している.

2.2 Weighted Decoding

Weighted Decoding(以下 WD と呼ぶ) は条件付き文生成で提案されたアプローチで, ある属性 a で条件付けられた文 X の生成 $P(X|a)$ を目的としている. WD では, ベイズの定理 $P(X|a) \propto P(X)P(a|X)$ により,$P(X|a)$ が $P(X)$ と $P(a|X)$ に分解できることを利用する. Liu ら [6] は, 望ましい属性 a のデータで学習した言語モデル $M+$ と望ましくない属性 $\bar{a}$ のデータで学習した言語モデル M^- を導入し, 2 つの言語モデル M^+ と M^- の対照性を利用して擬似的な $P(a|X)$ を計算している.

2.3 本研究の位置づけ

上述のように, 口調に関するスタイル変換研究は少なくどのような手法が有効であるかは分かっていない. また, 特定の口調を持つスタイル文は収集が難しく, 少資源であるという課題が存在する. そのため, 本研究では利用可能なデータに基づいて様々な状況を設定し, 各々の状況設定において口調変換に有効なスタイル変換手法の検討を行う. 特に, 単一のスタイルデータで学習可能な手法に焦点を当て, 教師あり手法, 逆翻訳による擬似対訳データ生成手法, WD をスタイル変換に応用する手法, ChatGPT[1]を用いた Few-shot 学習によるスタイル変換手法について比較検討を行う.

3 口調スタイル変換の定義とデータセット作成

3.1 口調スタイル変換の定義

本研究では口調スタイルを “特定の発話者を想起する特徴的な言葉遣い” と定義し, 方言や役割語などを対象とする. また, 一般的な話し言葉をノーマルスタイルとみなし, 口調スタイル変換を “ノーマルスタイル文から特定の口調スタイル文への書き換え” とする. 以降, 変換前のノーマルスタイル文を原文, 変換後の文を口調スタイル文, 変換前後のペアの集合を対訳データと呼ぶ.

[1]https://openai.com/chatgpt

3.2 口調スタイルデータセットの作成

日本語の口調制御に関する先行研究では, 独自の非公開データセットを使用しているものが多い. 本研究では公開データセットと口調変換ツールを活用し, 一般に再現可能なデータセットを新たに作成した.

3.2.1 作成方法

特定の口調を持った文を作成するため,Web 上で公開されている口調変換ツール[2] [3] [4]を利用した. ツールに入力する原文データには JPersonaChat データセット [2] を用いた. これは話者が自身の特徴を表すプロフィール文を設定し, プロフィールに基づいたワーカー同士の会話を収集したデータセットである. 多様な口調の文が含まれており, 入力の原文に対してロバストな口調スタイル変換器の検討を行うことができると考える.

JPersonaChat の各文に正規化と分割の前処理を行い約 10 万文を得た. この原文をツールで変換し, 口調スタイル文として「大阪弁」,「博多弁」,「お嬢様口調」,「廓口調」を作成した. 以降, 原文と口調スタイル文からなるデータセットを「口調スタイルデータセット」と呼ぶ. 表 1 に「良い天気なので、わたしは買い物に行きます。」という原文に対する口調スタイル文を示す.

表 1: 例文に対して作成された各口調スタイル文.

	例文
原文	良い天気なので、わたしは買い物に行きます。
大阪弁	ええ天気やさかい、うちは買い物に行く。
博多弁	良か天気やけん、うちは買い物に行く。
お嬢様口調	良いお天気なので、わたくしは買い物に行きますわ。
廓口調	良い天気なので、わっちは買い物に行きんす。

3.2.2 口調スタイルデータセットの分析

生成した口調スタイル文の中には, ツールで変換されなかった文が一定数含まれていた. ツールで変換できた文を「変換文」, 変換されなかった文を「無変換文」と呼ぶ. 無変換文は (i) 変換せずとも目的の口調スタイルであるか, (ii) 変換すべきであるがツールの変換規則に当てはまらず変換できなかったか, のいずれかが原因だと考えられる. また, 変換文の中には人目で見て不自然な文が一定数含まれており, 例えば「忙しいですが」という文をお嬢様口調に変換すると「忙しいですわが」という文が生成された. しかし, 本研究では変換ツールが生成した文を目的の口調スタイルだとみなした.

[2]https://ojosama.jiro4989.com
[3]https://github.com/anmonite/expression_trans
[4]https://www.8toch.net/translate

3.2.3 実験に用いるデータ

各口調スタイル変換手法に用いるデータとして, 変換文 50000 文を学習データ,5000 文を評価データ,1000 文をテストデータとした. ただし, 各手法がツールよりもロバストに変換できるかを検証するため, 無変換文 1000 文をテスト用データに加え計 2000 文とした.

4 比較検討する口調変換手法

4.1 手法ごとの条件比較について

各手法の説明を行う前に, 手法ごとの条件の違いを表 2 にまとめる. ここで, 対訳データは口調スタイルデータセット内の対訳データを利用するかを表し, △は手法内で擬似的な対訳データを生成することを示す. 言語モデル数は各手法で学習する言語モデルの数を表している. また,WD 手法では翻訳器による逆翻訳データを利用し言語モデルを 3 つ学習する実験と, 逆翻訳データを利用せず言語モデルを 2 つ学習する実験をそれぞれ試行した. 詳細は節 4.4.3 の推論において説明する.

4.2 教師あり手法

以降で説明する教師なし手法に対するベースラインとして, 対訳データを用いた教師あり学習によるスタイル変換を試行する. 原文を入力として対訳の口調スタイル文が出力となるよう翻訳タスク形式で言語モデルを学習し, 口調スタイル変換器を生成する.

節 5 の評価実験では, 口調スタイルデータセットの対訳データを用いて T5 モデル [7] のファインチューニングを行った. 学習データ数 {100, 500, 1000, 5000, 10000, 50000} でそれぞれ実験を行い, 評価データ数は学習データ数の 10%, テストデータは同じものを用いた.

4.3 逆翻訳手法

逆翻訳手法では Prabhumoye ら [5] の研究に倣い, 逆翻訳を用いて口調スタイル文からスタイル除去を試みる. 口調スタイル文は方言や役割語を含む特性上, 通常の翻訳器では適切に翻訳できない恐れがあるため, ロバストな翻訳器として Web 上で公開されている高性能な翻訳器を利用する. こうした翻訳器では潜在表現を利用できないケースがあるため, 潜在表現を利用する先行研究とは異なり, 逆翻訳で得られる翻訳文をノーマルスタイル文とみなして擬似的な対訳データの生成を行う.

評価実験では, 口調スタイル文に逆翻訳を行うことで生成した疑似対訳データを用いて教師あり手法と同様の学習を行った. 翻訳器の性能によるスタイル変換の精度の違いを調べるため, 逆翻訳に用いる翻訳器として DeepL 翻訳[5]と Hugging Face で公開されている NLLB 翻訳モデル[6][8],OPUS-MT 翻訳モデル [9] を利用した. また, 逆翻訳の中間言語として英語を選択した.

4.4 Weighted Decoding 手法

WD 手法では, 条件付き文生成で提案された Liu らの WD アプローチをスタイル変換に応用する手法を提案する. まず WD アプローチについて説明し, その後スタイル変換への応用手法について説明を行う.

4.4.1 Weighted Decoding アプローチ

一般的な自然言語を学習した言語モデル M は, 入力文の単語列 $X = \{x_1, \cdots, x_n\}$ が与えられたとき, 確率 $P(X)$ を自己回帰的に式 (1) で計算する:

$$P(X) = \prod_{t=1}^{n} P(x_t|x_{<t}) = \prod_{t=1}^{n} softmax(z_t) \quad (1)$$

ここで,z_t は言語モデルで計算される t 番目の単語の logit である. 特定のスタイル a を条件として文生成を行う場合, Liu らはベイズの定理を用いて $P(X|a)$ を $P(X)P(a|X)$ に分解し, 望ましいスタイル a の文をモデル化する言語モデル M^+ と望ましくないスタイル $\bar{a}$ の文をモデル化する言語モデル M^- の確率を組み合わせ, 擬似的な $P(a|x_{\leq t})$ を式 (2) で計算している:

$$P(a|x_{\leq t}) = \left(\frac{P^+(x_t|x_{<t})}{P^-(x_t|x_{<t})}\right) \quad (2)$$

さらに,$P(a|x_{\leq i})$ を言語モデル M の出力と掛け合わせ, 生成文の t 番目の単語の確率は式 (3) で計算される:

$$\tilde{P}(x_t|x_{<t}) \propto P(x_t|x_{<t}) \left(\frac{P^+(x_t|x_{<t})}{P^-(x_t|x_{<t})}\right) \quad (3)$$

4.4.2 スタイル変換への応用

スタイル変換では, 入力文 I の意味を保ちつつスタイル a で条件付けるために,$P(X|a, I)$ をモデル化する必要がある. そこで言語モデル M の代わりにノイズ除去オートエンコーダ (以下 DAE と呼ぶ) を導入し, 式 (3) を式 (4) に書き換える.

$$\tilde{P}(x_t|x_{<t}) \propto P(x_t|x_{<t}, I) \left(\frac{P^+(x_t|x_{<t})}{P^-(x_t|x_{<t})}\right) \quad (4)$$

式 (4) において, DAE で計算される確率 $P(x_t|x_{<t}, I)$ は入力文 I と近い内容の文の出力を試み, 式 (2) の項は DAE の出力確率をスタイル a に近づける操作を行うと推測される.

[5]https://www.deepl.com/translator
[6]https://huggingface.co/facebook/nllb-200-distilled-600M

表 2: 各手法ごとの比較項目.

比較項目	教師あり手法	逆翻訳手法	WD 手法	ChatGPT 手法
対訳データ	○	△	×	×
学習データ数	100〜50000	100〜50000	50000	10〜100
外部資源の利用	×	翻訳器	翻訳器/×	ChatGPT
言語モデル数	1	1	3/2	0

4.4.3 各モデルの学習および推論方法

DAE 口調スタイル文に対してノイズを付与し, ノイズが付与された口調スタイル文から元の口調スタイル文を復元するように口調スタイルごとに T5 モデルの学習を行った. 入力文に加えるノイズとして, 入力文の単語を一定確率で削除し (本研究では削除確率を 0.1 とした), その後入力文の単語列を無作為に並び替えた.

言語モデル M^+・M^- 先行研究に倣い, 言語モデル M^+・M^- として GPT[10] ベースのモデルを利用する. しかし, 式 (4) を計算する際に全ての言語モデルの語彙が統一されている必要があるため, GPT モデルの Tokenizer と単語埋め込み層 $W_{GPT} \in \mathbb{R}^{V_{GPT} \times 1024}$ の代わりに T5 モデルの Tokenizer と単語埋め込み層 $W_{T5} \in \mathbb{R}^{V_{T5} \times 2048}$ を利用した. ここで V は各モデルの語彙サイズを示す. また,T5 モデルと GPT モデルの単語埋め込み層の出力サイズが異なるため, 線形層 $W \in \mathbb{R}^{2048 \times 1024}$ を追加した. 言語モデル M^+ として GPT モデルを各口調スタイル文でファインチューニングしたものを用い, 言語モデル M^- には逆翻訳手法で生成した DeepL 翻訳による逆翻訳文でファインチューニングしたものを用いた.

推論 推論では, 式 (4) の計算を対数上で行い, DAE の重み付けとして, 出力 logit に対して temperature(T) を用いて式 (5) のように確率分布の操作を行った.

$$P(x_t) = \frac{exp(z_{t,i}/T)}{\Sigma_{j=1}^{|V|} exp(z_{t,j}/T)} \tag{5}$$

ここで,i と j は語彙 V における i(または j) 番目の単語を表す. T が 1 より小さいほど $P(x_t)$ における各単語の確率の差が大きくなり DAE の重みが大きくなる. また, 言語モデル M^+ と M^- の操作によって入力文 I の内容から遠い単語が選択されることを防ぐため, DAE の logit z_t に対して top k サンプリングを行い, 確率上位 k 個以外の単語の確率を 1×10^{-100} とした.

さらに追加の設定として, 言語モデル M^- を用いず DAE と言語モデル M^+ のみを用いた実験を行った. 2 つの実験を区別するため,M^+ と M^- を用いる実験を「M^+・M^- 実験」, 追加実験を「M^+ only 実験」と表記する.

4.5 ChatGPT 手法

大規模言語モデルはプロンプトによりモデルの応答スタイルを制御することが可能であり, 特に ChatGPT はその能力が高いことが知られている. そこで,ChatGPT を用いた Few-shot での口調変換を試行する.

評価実験では,OpenAI API の gpt-4-1106-preview モデルを利用した. プロンプトで例示する口調スタイル文は学習データからランダムに {10,50,100} 文抽出し, 各 shot 数を試行した. この際, モデルの出力トークン上限が 4k となっており, テストデータ全文の実験には多くの試行回数が必要であった. そのため, テストデータの変換文と無変換文を 100 文ずつ抽出した小規模データを用いて実験を行った. ChatGPT に与えたプロンプトを図 1 に示す. 赤色部分には与えるデータに応じて適

```
あなたは「{口調スタイル名}」で喋り、以下のサンプル文で示すような口調で話します。
<サンプル文>
{1: サンプル文}
…
{n: サンプル文}

以下のテスト文について、「{口調スタイル名}」の文に書き換えてください。
ただし、文の内容はできる限り保ち、文の変更は必要最低限にしてください。書き換えた文のみをテキスト番号とともに出力してください。
出力フォーマットは以下の通りです:
<出力文>
1: [書き換えた文]
…
100: [書き換えた文]

<テスト文>
{1: テスト文}
…
{100: テスト文}

<出力文>
```

図 1: ChatGPT に入力するプロンプトのテンプレート.

切な文が挿入され, { 口調スタイル名 } には例示する口調スタイル名が, { サンプル文 } には例示する口調スタイル文が,{ テスト文 } にはテストデータの文が入る.

5 評価実験

本実験では, テストデータの原文に対して各手法で口調スタイル変換を行い, 以下の評価指標を用いて各手法の口調スタイル変換性能を調べる.

5.1 評価指標

テキストスタイル変換では (i) スタイル変換精度,(ii) 内容の保持,(iii) 文の自然さ, の 3 つの指標で評価されることが望まれる [11].

スタイル変換精度 スタイル変換精度では, スタイル変換器の出力文が目的のスタイルであるかを測定する. 口調スタイル測定器として BERT[12] ベースの分類器を利用し, 分類結果の accuracy 値 (以下 ACC と呼ぶ) を算出した. 学習データの原文と口調スタイル文を用いて事前学習済みモデル[7]に対しバッチサイズ 100, 学習率 5×10^{-5},epoch 数 3 でファインチューニングを行った.

内容の保持 内容の保持では, スタイル変換前後の文の内容が同じかを測定する. 口調スタイル変換前後の文に対し SentenceBERT[13] を用いて文ベクトルを生成し, cos 類似度 (以下 SIM と呼ぶ) を算出した. 学習データの対訳データを用いて事前学習済みモデル[8]にバッチサイズ 128, 学習率 2×10^{-5},epoch 数 1 でファインチューニングを行い, 目的関数に MultipleNegativesRankingLoss [9]を用いた.

文の自然さ 文の自然さでは, スタイル変換器の出力文が目的スタイル下で文法的に自然であるかを測定する. GPT ベースの言語モデルで perplexity 値 (以下 PPL と呼ぶ) を算出した. 学習データの口調スタイル文を用いて事前学習済みモデル[10]に対しバッチサイズ 200, 学習率 5×10^{-4},epoch 数 10 で,LoRA 手法によりファインチューニングを行った.

5.1.1 各評価モデルの検証

各評価モデルが口調スタイル変換の評価能力を有しているか検証するため, テストデータおよび 2 種類のダミーデータで評価を行った. 表 3 に評価結果を示す. input copy はテストデータの原文をそのまま出力文とした際の評価結果を示し, different pair はテストデータの i 番目の原文と i+1 番目の口調スタイル文のペアの評価結果を示している. テストデータでは全ての指標で高い精度を示した一方で, input copy では ACC が 0.0053 と非常に低く,PPL もテストデータより高い値となった. また,different pair では対訳関係のない原文と口調スタイル文に対して, SIM が 0.0217 と類似度が非常に低いことを示しており, 各評価モデルが適切に評価できていることが分かる.

表 3: テストデータおよびダミーデータによる評価モデルの検証結果.

	変換文			無変換文		
	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
テストデータ	0.9823	0.9496	12.5754	0.1550	1.0000	23.8534
input copy	0.0053	1.0000	36.6903	0.1550	1.0000	23.8534
different pair	0.5686	0.0217	18.2110	-	-	-

5.2 事前学習済みモデルと学習設定

本実験では, 事前学習済み言語モデルとして T5 モデル[11]と GPT モデル[12]を用いた. 各モデルのファインチューニングには LoRA[14] 手法を利用し, バッチサイズ 200, 学習率 5×10^{-4}, epoch 数は step 数が 10000 になるよう学習データ数に応じて設定し,early_stopping を 3 とした. また, 推論には貪欲法を用いた.

5.3 実験結果

5.3.1 教師あり手法

教師あり手法の評価結果を表 4 に示す. 「変換文」において 3 つの評価指標全てで非常に高い精度となり, 学習データ数を減らすと SIM はあまり変わらず ACC と PPL の精度のみが低下した. また「無変換文」では,SIM が非常に高くなっている一方,ACC と PPL の精度が大きく低下した. ただし, 学習データ数を減らしていくと,SIM と PPL の精度を大きく低下させることなく ACC が上昇することが確認できる.

表 4: 教師あり手法の評価結果.

	変換文			無変換文		
データ数	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
50000	0.9800	0.9505	12.4807	0.1955	0.9982	23.4522
10000	0.9783	0.9518	12.5632	0.2155	0.9976	23.7861
5000	0.9725	0.9529	12.8685	0.2400	0.9961	23.7418
1000	0.9188	0.9571	14.8272	0.2963	0.9920	23.7263
500	0.8903	0.9585	15.6864	0.3100	0.9900	24.4895
100	0.7610	0.9608	21.0600	0.2860	0.9866	24.7192

[7]https://huggingface.co/line-corporation/line-distilbert-base-japanese
[8]https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2
[9]https://www.sbert.net/docs/package_reference/losses.html
[10]https://huggingface.co/rinna/japanese-gpt-neox-3.6b
[11]https://huggingface.co/retrieva-jp/t5-xl
[12]https://huggingface.co/rinna/japanese-gpt-neox-small

5.3.2 逆翻訳手法

各翻訳モデルの結果を表5に示す．「変換文」において ACC は 0.98 前後,PPL は 10 以下となっており, 教師あり手法と同等以上の結果となった.SIM の精度は翻訳モデルによって大きく異なっており, 一番高い DeepL で 0.8826, 一番低い OPUS-MT で 0.6195 となった．また,「無変換文」でも翻訳モデルによらず ACC と PPL が教師あり手法の結果を上回り, 教師あり手法に比べて「無変換文」でも変換できていることが分かる．

表 5: 逆翻訳手法における各翻訳モデルの評価結果.

	変換文			無変換文		
翻訳モデル	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
DeepL	**0.9840**	**0.8826**	9.6935	0.7878	**0.8937**	12.3695
NLLB	0.9778	0.7948	8.3483	0.8565	0.7854	9.1317
OPUS-MT	0.9740	0.6195	**6.5307**	**0.8945**	0.5994	**6.8972**

5.3.3 Weighted Decoding 手法

M^+・M^- 実験　WD 手法における「M^+・M^- 実験」(節 4.4.3 参照) の評価結果を表 6 と表 7 に示す．ここで T は temperature の値,k は topk サンプリングの値である． 表 6 から,T の値が小さいほど DAE の出力の重みが強く, SIM が高くなる代わりに ACC が低くなることが分かる．また, 他手法では ACC と PPL の精度にはある程度の相関があったが, WD 手法では ACC の精度が上がっても PPL の精度が下がっている．

表 6: M^+・M^- 実験の T ごとの評価結果 (k=100).

	変換文			無変換文		
T	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
0.3	0.5618	**0.9592**	23.5437	0.3923	**0.9809**	**20.9174**
0.4	0.7023	0.9395	**22.8568**	0.4680	0.9707	21.4520
0.5	0.7855	0.8892	29.1263	0.5338	0.9109	31.4880
0.6	0.8525	0.8332	41.7806	0.6380	0.8302	50.1507
0.7	**0.8943**	0.7961	50.5259	**0.7070**	0.7848	60.4794

表 7: M^+・M^- 実験の k ごとの評価結果 (T=0.5).

	変換文			無変換文		
k	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
1	0.1668	**0.9875**	30.2315	0.2168	**0.9885**	**23.1108**
10	0.7668	0.8957	**24.9705**	0.5190	0.9175	28.0899
100	0.7855	0.8892	29.1263	0.5338	0.9109	31.4880
1000	**0.7868**	0.8887	29.2136	**0.5345**	0.9105	31.4250
10000	**0.7868**	0.8887	29.2136	**0.5345**	0.9105	31.4250

次に, 表 7 から k の値が小さいほど言語モデル M^+ と M^- が操作できる語彙が減少し, SIM の精度が上がり ACC が下がることが分かる．ただし,T の値の操作ほど各指標の精度に影響を与えていない．また,k=1 は DAE のみで出力を生成している状態を示し, 言語モデル M^+ と M^- の操作を受けないため ACC が低くなった．

M^+ only 実験　表 8 に「M^+ only 実験」(節 4.4.3 参照) の評価結果を示す．言語モデル M^- を用いておらず言語モデル M^+ による影響が強くなるため, T の値を高く設定して DAE とのバランスをとっている．また,ACC と PPL の精度に相関が出ており, 総合的な評価では M^+・M^- 実験よりも高い変換精度となった．

表 8: M^+ only 実験の T ごとの評価結果 (k=100).

	変換文			無変換文		
T	ACC ↑	SIM ↑	PPL ↓	ACC ↑	SIM ↑	PPL ↓
0.8	0.7667	**0.8759**	15.1983	0.4400	**0.9181**	15.8812
0.9	0.8108	0.8569	14.4806	0.4935	0.8952	15.3171
1.0	0.8505	0.8379	13.9671	0.5670	0.8676	14.8000
1.1	0.9098	0.8026	13.7211	0.7028	0.8190	14.7624
1.2	**0.9313**	0.7771	**13.3558**	**0.7495**	0.7853	**14.1810**

5.4 口調スタイルごとの標準偏差

口調スタイルごとの評価結果に偏りがないか調べるため, 各指標の標準偏差を算出した．表 9 に標準偏差の値を示す．逆翻訳手法では教師あり手法と同等の標準偏差に収まっている一方で, WD 手法の ACC および ChatGPT 手法の全ての指標で標準偏差が大きくなっており, スタイルごとの性能に偏りがあることが分かる．

表 9: 各手法の評価値の標準偏差.

	変換文			無変換文		
手法	ACC	SIM	PPL	ACC	SIM	PPL
教師あり	0.0089	0.0159	1.5458	0.1250	0.0003	3.7138
逆翻訳	0.0117	0.0116	0.9766	0.0948	0.0106	1.9567
WD	0.0762	0.0266	2.3148	0.1588	0.0294	1.5969
ChatGPT	0.0658	0.0418	4.8774	0.0580	0.0418	10.9406

5.5 学習データ数による比較

データ数 50000 と 100 における各手法の評価結果について表 10 と表 11 にまとめる．ここで, 逆翻訳手法は DeepL 翻訳の結果を, WD 手法は「M^+ only 実験」(T=1.1, k=100) の結果を記載している． 表 10 よりデータ数 50000 では逆翻訳手法が ACC と PPL で最も高い評価値となり, SIM は教師あり手法から 0.07 程度低い値で, 総合的に教師あり手法に匹敵する結果となった．また, 表 11 よりデータ数 100 では教師あり手法と逆翻訳手法で ACC の評価値が大幅に低下しており, ChatGPT 手法が総合的に最も高い性能となった．

表 10: 学習データ数 50000 における各手法の評価結果.

	変換文			無変換文		
手法	ACC	SIM	PPL	ACC	SIM	PPL
教師あり	0.9800	**0.9505**	12.4807	0.1955	**0.9982**	23.4522
逆翻訳	**0.9840**	0.8826	**9.6935**	**0.7878**	0.8937	**12.3695**
WD	0.9098	0.8026	13.7211	0.7028	0.8190	14.7624
ChatGPT	-	-	-	-	-	-

表 11: 学習データ数 100 における各手法の評価結果.

	変換文			無変換文		
手法	ACC	SIM	PPL	ACC	SIM	PPL
教師あり	0.7610	**0.9608**	**21.0600**	0.2860	**0.9866**	24.7192
逆翻訳	0.5903	0.9034	23.6577	0.5028	0.9376	**22.3865**
WD	-	-	-	-	-	-
ChatGPT	**0.9050**	0.8810	23.3524	**0.8675**	0.8589	29.8910

6 考察

6.1 無変換文に対する変換性能

表 4 から, 教師あり手法の「無変換文」では目的の口調スタイルでない入力文が変換されずに出力されている状態だと考えられる. セクション 3.2.3 において学習データに無変換文を含まなかったにも関わらず, 無変換を変換モデルが学習してしまっていることが伺える. これは, 変換文内に含まれる無変換箇所を変換モデルが過学習してしまったためだと考えられる. そのため, 学習データ数を減らすことで過学習が抑えることができ,ACC が上昇したのではないかと推察できる.

一方, 全ての教師なし手法で「無変換文」でも一定の ACC・PPL 精度が出ており, 教師あり手法よりもロバストに変換できていることが分かる. 対訳データを用いる逆翻訳手法でもロバストな変換ができているのは, 逆翻訳で得られる文が多様性を持ち, 様々な変換パターンが学習データに含まれたためだと考えられる. 例として, 大阪弁の変換ツールは「いい天気だ」を「ええ天気や」に変換するが, 「いい天気です」は「ええ天気です」と「です」の部分が変換されず, 「〜です」から「〜や」への変換パターンが学習データに含まれない. 一方, 逆翻訳では「〜や」が「〜だ」や「〜です」など様々な形に逆翻訳され, 多様なパターンを学習データに含むことができたのではないかと考えられる.

6.2 PPL が高く算出された手法とその原因

WD 手法では, 言語モデル M^+ と M^- の対照性を利用して $P(a|x)$ を計算しており, 言語モデル M^- を用いない場合, スタイルに関わらず言語モデル M^+ の学習データに頻出する単語にバイアスがかかる恐れがある. しかし, 文の自然さの評価モデルも言語モデル M^+ と M^- と同様の言語分布を持つデータで学習しているため, PPL が低く算出される単語の生成を言語モデル M^- が遠ざけてしまっていると考えられる. そのため,T の値を大きくすると言語モデル M^- の影響が強くなり ACC の上昇とは反対に PPL 値が悪化し, 一方で言語モデル M^- を用いない「M^+ only 実験」では ACC の精度と PPL の精度が共に上昇していると推察される.

ChatGPT 手法では,ChatGPT が事前学習したデータと本実験で扱った口調スタイルデータの言語分布に乖離があったため PPL が大きくなったと考えられる. 具体的には, セクション 3.2.2 で述べたように口調スタイル文には変換ツールの規則不備に起因する不自然な文が含まれており, 文法の自然さの評価モデルも変換ツールが生成した言語分布を学習している. ChatGPT は事前学習データに含まれる大阪弁や博多弁などの知識も生成に活用していると思われるが, 変換ツールが生成した口調スタイルがそうした一般的な方言・口調と乖離があったのではないかと考えられる.

6.3 各教師なし手法の詳細について

逆翻訳手法 表 10 より, データが十分に用意できる場合, 逆翻訳手法は他手法よりも高いスタイル変換性能を示しており, 逆翻訳が口調スタイル変換に有効であることが伺える. ただし, 翻訳モデルの性能によって文の内容保持性能が大きく左右されるため, 逆翻訳に用いる翻訳モデルの選択が重要となり, 翻訳が困難な口調スタイルを扱う場合は精度が低下する可能性がある. これは, ロバストな翻訳モデルでは口調スタイル文を適切に翻訳することができ逆翻訳文が擬似的な対訳文として上手く機能した一方で, 口調スタイル文を上手く翻訳できないモデルでは不完全な対訳データが生成され, 入力文との意味的類似性を考慮せず文生成を行う変換モデルが学習されるためだと考えられる.

Weighted Decoding 手法 表 10 から,WD 手法は他手法に僅かに及ばない結果となった. SIM の精度が低い原因として,DAE が削除された単語を復元するように訓練されているため, 元の文で省略されている主語や文末などの単語を補完して文を生成してしまっていることなどが考えられる. 一方,WD 手法では temperature と top k の値を操作することで生成を制御可能な利点がある. スタイル変換では変換精度と文の内容保持との間にトレードオフが存在するが, 上記の値を操作することでトレードオフのバランスを取ることができるため, 実課題における可用性が高いと考えられる.

ChatGPT 手法 表 11 より, データ数が非常に少数の場合は ChatGPT 手法が最も高い変換性能を示した. また, 表 10 が示すデータ数 50000 における各手法の結果

と比較しても競争力のある変換性能を発揮した．一方で, 表 9 の結果より扱う口調スタイルごとの性能の偏りが他手法より大きいことも明らかとなった．これは,6.2 で述べたように口調スタイル文と ChatGPT が事前学習した大阪弁や博多弁などの文の間に乖離が存在したからだと考えられる．実際に, 他の教師なし手法では各スタイルの中でも比較的 ACC が高く出ている大阪弁について,ChatGPT 手法では非常に低い値となった．

7 おわりに

対話システムの口調制御に向け, 方言や役割語を対象とした口調スタイル変換について, 複数の教師なしテキストスタイル変換手法の比較検討を行った．実験の結果, 全ての教師なし手法で教師あり手法よりもロバストな変換性能を示した．データ数が豊富な場合には, 逆翻訳手法が教師あり手法に最も近い高い性能を示したが, 翻訳器によって変換性能が左右されるため, 扱う口調をロバストに翻訳できる翻訳器が重要となる．また, データ数が限られる場合には ChatGPT を用いた Few-shot 学習が最も高い変換性能を示したが, 口調による性能差が大きいという欠点も確認された．本研究で提案した WD 手法は他手法にやや及ばなかったものの, 従来の WD アプローチに DAE を組み合わせることでスタイル変換に活用可能だと判明し, パラメータ操作により変換精度と内容保持とのトレードオフを調整できる点で有用性を示すことができた．

本研究で扱わなかった口調スタイル変換として, 口調スタイル文からノーマルスタイル文への変換や, 口調スタイル文から別の口調スタイル文への変換などが挙げられる．これらの変換について今後の課題としたい．

参考文献

[1] 赤間怜奈ほか. 転移学習を用いた対話応答のスタイル制御. 言語処理学会第 23 回年次大会発表論文集, pp. 338–341, 2017.

[2] Hiroaki Sugiyama, et al. Empirical analysis of training strategies of transformer-based japanese chit-chat systems, 2021.

[3] 谷川晃大ほか. 変分オートエンコーダと注意機構を用いた発話文のキャラクタ性変換. 人工知能学会全国大会論文集 第 32 回全国大会, p. 4G201, 2018.

[4] 江崎拓哉ほか. Flow-base モデルを用いた文のスタイル変換. 第 12 回データ工学と情報マネジメントに関するフォーラム, No. F7-1, 2020.

[5] Shrimai Prabhumoye, et al. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 866–876, 2018.

[6] Alisa Liu, et al. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 6691–6706, 2021.

[7] Colin Raffel, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, Vol. 21, No. 1, pp. 5485–5551, 2020.

[8] NLLB Team, et al. No language left behind: Scaling human-centered machine translation. 2022.

[9] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation*, 2020.

[10] Alec Radford, et al. Improving language understanding by generative pre-training. 2018.

[11] Remi Mir, et al. Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 495–504, 2019.

[12] Jacob Devlin, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, pp. 4171–4186, 2019.

[13] Nils Reimers, et al. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019.

[14] Edward J Hu, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.