E-025

スタイル変換による雑談対話システムへのキャラクタ性の付与
Dialogue System Characterization by Style Transfer

近藤 里咲† 梶川 怜恩† 梶原 智之† 二宮 崇†
Risa Kondo Reon Kajikawa Tomoyuki Kajiwara Takashi Ninomiya

1. はじめに

近年，人間と自然に会話できる雑談対話システム [1-3] が開発されつつあり，生成される応答の妥当性や流暢性を超えて，キャラクタ性などの生成スタイルの制御 [4-6] に関心が高まっている．発話のキャラクタ性は文末表現や音変化表現などの言い回しによって特徴づけられ [7]，これらの言語表現を制御することで雑談対話システムがユーザに与える印象を変化させられる．会話相手として親しみやすく，ユーザに愛される雑談対話システムを実現するために，りんな[1]やなりきり AI プロジェクト[2]など，雑談対話システムへの特定のキャラクタ性の付与が試みられている．

雑談対話システムにキャラクタ性を付与するために，先行研究では転移学習 [4] や強化学習 [6] の手法を採用している．つまり，汎用的な雑談対話モデルを訓練した後，一貫したキャラクタ性を持つコーパスを用いて，教師あり学習または強化学習の枠組みで fine-tuning を実施している．これらの手法では，比較的小規模の発話-応答パラレルコーパスまたは特定のキャラクターに関する発話コーパスのみを用いて雑談対話システムにキャラクタ性を付与できる．しかし，ChatGPT[3]のような通常の環境で訓練できない[4] サイズの大規模モデルやブラックボックスシステムに対しては，fine-tuning の適用は難しい．

本研究では，所与の雑談対話システムに対して fine-tuning なしでキャラクタ性を付与するために，応答生成とスタイル変換を組み合わせたパイプラインモデルを提案する．提案手法では，ブラックボックスの応答生成モデルを想定し，任意のモデルが生成できる応答文に対してスタイル変換によってキャラクタ性を付与する．スタイル変換モデルは，対象キャラクターの発話に対して折り返し翻訳によってキャラクタ性を除去し，それを復元する訓練を通して構築する．そのため，我々のスタイル変換モデルは大規模な発話-応答パラレルコーパスを必要とせず，対象キャラクターの発話コーパスのみを用いて訓練できる．任意の応答生成モデルを扱える本手法は，ChatGPT などの大規模言語モデルに基づく高性能な応答生成の恩恵を受けられる．

複数のご当地キャラクターに関する対話データにおける実験の結果，自動評価と人手評価の両方において，スタイル変換によって応答のキャラクタ性を改善できた．人手評価では，ChatGPT は流暢性と妥当性において優れているものの，提案手法によってキャラクタ性を改善でき，総合評価としては応答生成とスタイル変換のパイプラインモデルが最高性能を達成した．我々が提案するスタイル変換モデルは効率的に訓練できるため，数百件程度の発話さえ入手できれば，大規模な事前訓練を経た応答生成モデルの恩恵を受けつつ，キャラクタ性を持つ雑談対話を実現できる．

2. 関連研究

2.1 雑談対話システム

対話システムは，タスク指向型と非タスク指向型に大別できる [10]．タスク指向型は，チケットの予約など，特定のタスクを遂行することを目的とする対話システムである．非タスク指向型は，特定のタスクを想定しない対話システムであり，雑談対話システムとも呼ばれる．雑談対話システムのアプローチには，ルールベース [11]，用例ベース [12]，生成ベース [13] の大きく 3 種類がある．ルールベースの手法は，人手で応答文を作成するため，制御性が高い一方で幅広い話題に対応するためには多くのコストが必要となる．用例ベースの手法は，既存の発話-応答パラレルコーパスから応答文を抽出するため，構築コストが低い一方で制御性や柔軟性に欠ける．生成ベースの手法は，幅広い話題に対して流暢な応答が可能であり，特に Transformer [14] を大規模に事前訓練した生成ベースのモデル [1, 2] はルールベースや用例ベースよりも高い性能を達成するため，現在の主流のアプローチとなっている．

日本語の雑談対話においても，Transformer に基づく事前訓練モデル [15, 16] が活用されている．NTT の対話モデル [3] は，Twitter の投稿-返信対を用いて事前訓練された 16 億パラメタの大規模な Encoder-Decoder 型 Transformer を，対話コーパス [17] 上で fine-tuning したものである．rinna の対話モデルは，CC-100 [18] などの生コーパス上で言語モデリングの事前訓練を経た 36 億パラメタの大規模な Decoder 型 Transformer を，FLAN [19] などの instruction-tuning 用コーパスの和訳を用いて fine-tuning したものである．本研究では，これらの生成ベースの雑談対話システムを対象に，キャラクタ性の付与に取り組む．

2.2 雑談対話システムへのキャラクタ性の付与

雑談対話システムにキャラクタ性を付与する先行研究にはルールベースの手法 [5] と fine-tuning ベースの手法 [4, 6] がある．前者は訓練を必要としない一方で，キャラクターごとにルールを作成するための多くのコストが必要となる．後者はデータ駆動の効果的な手法であるが，先述のように，巨大なモデルやブラックボックスシステムなどの fine-tuning できないモデルに対しては適用できない．

Fine-tuning ベースの手法のうち，Akama ら [4] は，特定のキャラクターに限定しない大規模な発話-応答パラレル

† 愛媛大学 Ehime University
[1] https://www.rinna.jp/
[2] https://narikiri-qa.jp/
[3] https://chat.openai.com/ (API は gpt-3.5-turbo モデル)
[4] 訓練の効率化に関する研究 [8] も進んではいるが，この技術を用いても現在の通常の計算環境では GPT-3 [9] などの大規模言語モデルを fine-tuning することは難しい．

表 1：折り返し翻訳によるキャラクタ性の除去の例

元の発話	無事に届いてよかったにゃべ 今週もがんばろうにゃべー!!
英語への翻訳	I'm glad it arrived safely Let's do our best this week too!!
折り返し翻訳	無事に届いてよかったです 今週も頑張りましょう！！

コーパスを用いて応答生成モデルを訓練した後，特定のキャラクターによる応答で構成される発話-応答パラレルコーパス上で追加の訓練を行う転移学習の手法を提案した．また，清水ら [6] は，GPT-2 [15] の事前訓練済み言語モデルを特定のキャラクターに限定しない大規模な発話-応答パラレルコーパス上で fine-tuning して応答生成モデルを構築した後，さらに強化学習によってキャラクタ性を付与している．これらの先行研究とは対照的に，本研究では，fine-tuning できないモデルを含む任意の応答生成モデルに対してキャラクタ性の付与に取り組む．

2.3 折り返し翻訳と言い換え

英語の言い換え文対の収集において，折り返し翻訳に基づく手法 [20] が研究されている．こうして収集された言い換えは，スタイル変換のために有用であることが報告 [21] されている．本研究でも，折り返し翻訳によって言い換えを獲得し，スタイル変換モデルの訓練に活用する．

3. 提案手法

応答生成モデルの fine-tuning なしでキャラクタ性を持つ雑談対話を実現するために，本研究では応答生成とスタイル変換のパイプラインモデルを提案する．提案手法では，応答生成モデルの出力文に対する後編集という位置付けでスタイル変換を適用し，応答にキャラクタ性を付与する．

3.1 疑似的なスタイル変換コーパスの作成方針

特定のキャラクターを想定しない一般的な発話文からキャラクタ性を持つ発話文へのスタイル変換を行うために，意味を保持しつつキャラクタ性の有無を変化させたパラレルコーパスを収集したい．しかし，このようなパラレルコーパスを大規模に収集可能なキャラクターは稀であるため，本研究では疑似的なパラレルコーパスを作成する．本研究では，Twitter などから大規模に発話を収集可能なキャラクターを想定し，スタイル変換モデルを訓練するための疑似的なパラレルコーパスを作成する．

2.3 節で述べた折り返し翻訳によって入力文の言い換えを生成できるという報告 [20, 21] および機械翻訳によってテキストから書き手に特有のスタイル情報を除去できるという報告 [22] に着想を得て，本研究では特定のキャラクターの発話から折り返し翻訳によってキャラクタ性を除去した言い換えを生成し，スタイル変換のための疑似的なパラレルコーパスを作成する．表 1 に，ご当地キャラクターであるキャベッツさんの Twitter 上の発話から，折り返し翻訳によってキャラクタ性を除去する例を示す．このように，折り返し翻訳したテキストと元の発話との対を，キャラクタ性の有無を変化させたパラレルコーパスとして用いることができる．そのため，折り返し翻訳のテキストから元の発話を復元する訓練を通して，入力テキストにキャラクタ性を付与するスタイル変換モデルを構築できる．

3.2 Twitter の発話からのスタイル変換モデルの学習

本研究では，対象キャラクターの Twitter 上の投稿テキストを用いて，スタイル変換のための疑似的なパラレルコーパスを作成する．Twitter の投稿には，通常の投稿・返信・引用・リツイートの 4 種類が含まれる．このうち，リツイートは他者による投稿であり，対象キャラクターの特徴を含まないため除外する．また，返信は発話-応答対として後述の評価用コーパスに用いるため，ここでは使用しない．その他の，通常の投稿および引用を用いて，スタイル変換のための疑似的なパラレルコーパスを作成する．ただし，引用については，引用されている他者の投稿は除外し，対象キャラクターの発話部分のみを抽出して用いる．

対象キャラクターの投稿からキャラクタ性を持たない言い換えを生成するために，折り返し翻訳を行う．ただし，複数文からなる投稿を翻訳する際に一部の内容が省略されてしまう場合が見られたため，投稿を 1 文ずつに文分割[5]して折り返し翻訳を行う．ここで，翻訳誤りを除外するために，文長が入力文字数の 0.5 倍未満または 2 倍以上の場合には，その文対をパラレルコーパスに含めない．このように作成された疑似的なパラレルコーパスを用いて BART [16] を fine-tuning し，キャラクタ性を付与するスタイル変換器を構築する．

3.3 応答生成モデルの後編集としてのスタイル変換

本研究では，応答生成モデルとして既存の訓練済みモデルに加えて，Twitter の投稿-返信対を用いて BART [16] を訓練するドメイン特化の対話モデルも扱う．提案手法では，所与の発話に対するキャラクタ性を持つ応答を生成するために，これらの応答生成モデルと 3.2 節で構築したスタイル変換モデルを組み合わせる．まず，任意の応答生成モデルに発話を入力し，特定のキャラクターを想定しない応答を生成する．そして，スタイル変換モデルにこの応答文を入力し，キャラクタ性を付与した応答文に変換する．

4. 評価実験

本研究では，一般人と区別可能なキャラクタ性を持ち，Twitter において 1 万件以上の投稿履歴を持つご当地キャラクターを対象に，評価実験を行う．所与の発話に対して，対象キャラクターになりきって応答する雑談対話システムを構築し，その性能の自動評価および人手評価を行う．

4.1 データセット

先述の対象キャラクターのうち，本実験への協力に同意を得た 4 キャラクター（オカザえもん，キャベッツさん，ちぃたん☆，レルヒさん）の Twitter データを用いて，スタイル変換および応答生成のデータセットを作成した．各ご当地キャラクターの発話例を表 2 に示す．

[5] https://github.com/megagonlabs/ginza

表 2：ご当地キャラクターの発話の例

オカザえもん	ありがとうでござる! ご飯に肉が乗ってるだけでござる	https://twitter.com/okazakiemon
キャベッツさん	今日は一緒に歌って踊れて楽しかったにゃべー♪ また遊ぼうにゃべね♪	https://twitter.com/kyabettsusan
ちぃたん☆	お友達と追いかけっこしましたっ☆ちぃたん☆ですっ☆	https://twitter.com/chiitan7407
レルヒさん	ぶろぐ 書イテタンヤ 秘密ニ シトケッテ 言ッタノニ	https://twitter.com/TheodorVonLerch

4.1.1 スタイル変換コーパス

スタイル変換のための疑似的なパラレルコーパスの作成において，前処理として以下の表現の正規化を行う．

- 半角カナを全角化
- 英数字などの ASCII 文字を半角小文字化
- メールアドレスを特殊トークン “<mail>” に置換
- メンション, URL, 文末ハッシュタグ, 絵文字の除去
- BART の語彙に含まれない記号を削除
- 改行を半角空白に置換し，連続する空白を 1 つに結合

ここで，記号の削除はスタイル変換モデルによって出力不可能な顔文字を除去することが目的である．具体的には，顔文字の検出性能が高い nagisa[6]を用いて発話を単語分割し，補助記号という品詞で顔文字を抽出した．この顔文字の中に，BART の語彙に含まれないトークンが存在する場合は当該顔文字を削除した．また，ハッシュタグの除去においては，発話を構成する単語としての機能を持つ文中のハッシュタグを削除すると文意を損なうため，文末に出現するハッシュタグのみを正規表現によって削除した．

前処理済みの投稿に対して，ノイズを避けるために以下のフィルタリングを実施した．

- 半分以上の文字が ASCII 文字で構成される投稿を削除
- Type Token Ratio が 0.2 未満である投稿を削除
- 4 単語以下または 100 単語以上の投稿を削除
- 他の投稿との Jaccard 係数が 0.8 を超える投稿を削除

最後のフィルタリングは，日付のみが異なるような定型表現を避けることが目的であり，単語単位での Jaccard 係数を求めた．なお，表 2 に示したように，レルヒさんの投稿は平仮名と片仮名が反転しているのが特徴である．この特徴のために，前処理およびフィルタリングの中で単語分割に失敗するため，前処理の前に平仮名と片仮名を反転させ，フィルタリングの後で再度反転させた．

前処理およびフィルタリングを経た発話に対して Google 翻訳[7]を用いた折り返し翻訳によって，キャラクタ性を除去した言い換えを生成し，スタイル変換のための疑似的なパラレルコーパスを作成した．なお，折り返し翻訳における中間言語には英語を使用した．訓練用・検証用・評価用に分割したスタイル変換コーパスの文対数を表 3 に示す．

4.1.2 応答生成コーパス

本節では，ドメイン特化の応答生成モデルを訓練するための発話-応答パラレルコーパスについて述べる．本コーパスは，2023 年 2 月 4 月から 3 月 25 日までの期間の Twitter 上の投稿-返信対から作成した．収集対象は日本語を含む投稿を行うアカウントに限定し，自身への返信は対象外とした．なお，本コーパスでは表 2 に示したご当地キャラクターの投稿は含まれないことに注意されたい．

表 3：データセットの文対数

	スタイル変換		
	訓練用	検証用	評価用
オカザえもん	28,000	269	268
キャベッツさん	2,000	234	233
ちぃたん☆	2,000	138	138
レルヒさん	62,000	163	163

投稿および返信のそれぞれに対して 4.1.1 節と同様の前処理およびフィルタリングを実施した．ただし，Jaccard 係数のみ変更を加えた．他の投稿との類似性ではなく，投稿と返信の類似性を評価し，閾値を 0.5 に設定した．なお，フィルタリングの際は，投稿または返信の少なくとも片方が削除の基準を満たす場合に，その文対を削除した．その結果，約 700 万文対のうち 500 万強の文対が残った．本実験では，このうち無作為に抽出した 500 万文対を訓練用，5,000 文対を検証用に使用した．

4.1.3 パイプライン評価用コーパス

本節では，キャラクタ性を持つ応答生成を評価するための発話-応答パラレルコーパスについて述べる．対象キャラクターによる Twitter 上の返信の投稿と，その返信元の投稿の対を用いて，本パラレルコーパスを作成する．ただし，対象キャラクター自身への返信は対象外とした．4.1.2 節と同様の前処理およびフィルタリングを行い．対象キャラクターの応答が重複する文対を削除した．収集した文対の中から，無作為抽出した 1,000 文対ずつを評価用に用いる．

4.2 実験設定

4.2.1 訓練の詳細と比較手法

スタイル変換モデルおよびドメイン特化の応答生成モデルには，事前訓練済みの BART[8] [16] を使用した．BART の fine-tuning においては，最適化関数として Adam [23] を使用し，バッチサイズを 2,048，ラベルスムージングを 0.2，ドロップアウト率を 0.3，学習率を 3e-5，Warmup ステップを 2,500 とし，検証用データにおけるクロスエントロピー損失が 32 エポック改善されなくなるまで訓練した．

また，その他の既存の応答生成モデルとして，2.1 節で紹介した NTT の対話モデル[9] [3]，rinna の対話モデル[10]，ChatGPT[3] も用いた．ChatGPT は API を用いて，付録 A.1 に示すゼロショットのプロンプトによって出力を得た．比較手法として，同じく付録に示す 3 ショットのプロンプトを用いて ChatGPT の応答生成にキャラクタ性を付与した．以降では，ゼロショットの ChatGPT と区別して，比較手法を ChatGPT（3-shot）と表記する．

[6] https://github.com/taishi-i/nagisa
[7] https://cloud.google.com/translate 2023/05/26-28 実行
[8] https://huggingface.co/ku-nlp/bart-large-japanese
[9] https://github.com/nttcslab/japanese-dialog-transformers
[10] https://huggingface.co/rinna/japanese-gpt-neox-3.6b-instruction-sft

表 4：スタイル変換の評価(BLEU)

	原文	BLEU
オカザえもん	29.19	**62.07**
キャベッツさん	26.73	**46.00**
ちぃたん☆	26.32	**59.16**
レルヒさん	12.15	**32.52**

表 5：パイプライン評価（左がスタイル変換前 BLEU，右がスタイル変換後 BLEU）

	オカザえもん	キャベッツさん	ちぃたん☆	レルヒさん
NTT	0.44 → 3.09	0.72 → 4.27	0.37 → 3.87	0.06 → 0.29
rinna	0.55 → 2.37	0.89 → 4.50	0.43 → 4.34	0.11 → 0.64
ChatGPT	0.68 → 4.68	1.50 → 7.42	0.43 → 5.00	0.07 → 0.89
BART	1.89 → **5.40**	3.46 → **8.70**	0.84 → **5.04**	0.44 → **1.98**
ChatGPT（3-shot）	2.65	4.27	3.46	0.18

4.2.2 評価指標

本実験では，自動評価と人手評価の両方で生成文の品質を評価した．自動評価においては，スタイル変換と応答生成の両方で BLEU [24] を用いた．人手評価においては，本実験で扱うご当地キャラクターのうち，最も Twitter のフォロワー数が多いちぃたん☆を対象に，キャラクタ性・流暢性・妥当性の 3 つの観点から応答生成を評価した．具体的には，日本語を母語とする大学生 3 名が，無作為抽出された 100 件の応答文を 5 段階評価した．また，全ての評価観点において評価者3名の平均点が4.0以上である応答を「良い応答」と定義し，モデルごとの「良い応答」の割合に基づいて総合評価を行った．

人手評価における評価者間の一致度を重み付きカッパ値を用いて求めたところ，キャラクタ性は 0.63-0.82，流暢性は 0.27-0.52，妥当性は 0.55-0.67 であった．流暢性については一部の評価者間で充分な一致とは言えないものの，機械翻訳の人手評価においても評価者間の一致度は 0.25-0.39 との報告 [25] があるため，許容範囲内であると考えられる．

4.3 実験結果

4.3.1 自動評価

スタイル変換の内的評価の結果を表 4 に示す．ここで，原文という列には，対象キャラクターの発話とその折り返し翻訳の間の BLEU 値を掲載している．BART という列は，原文を提案手法によってスタイル変換した際の BLEU 値であり，スタイル変換によって全てのキャラクターに対してキャラクタ性を大幅に改善できることがわかる．

応答生成とスタイル変換によるパイプラインモデルに対する BLEU による自動評価の結果を表 5 に示す．4 つの応答生成モデルの全てにおいて，スタイル変換によってキャラクタ性を持つ応答生成の品質を大幅に改善できることがわかる．比較手法である ChatGPT の few-shot モデルと比べると，rinna モデルにオカザえもんのキャラクタ性を付与する場合を除いて，提案するパイプラインモデルが高い性能を示した．特に，ドメイン特化の BART ベースの応答生成モデルを用いると，キャラクターごとに比較手法よりも 1.58 から 4.43 ポイントの BLEU の改善が見られた．

4.3.2 人手評価

モデルごとの人手評価の平均値と総合評価の結果を表 6 に示す．キャラクタ性では BART の応答生成モデルにスタイル変換を組み合わせた提案モデル，流暢性および妥当性では比較手法である ChatGPT の few-shot モデルが最も高い評価を得た．ChatGPT の few-shot モデルは自然な応答生成ができるものの，キャラクタ性が不充分な場合が多く，全ての評価観点において 4.0 点以上を獲得できる例は多くなかった．上段 2 行の比較からもわかるように，スタイル変換の処理を加えることによって，流暢性および妥当性をわずかに損ないつつもキャラクタ性を大幅に改善できる．そのため，3 つの評価観点を組み合わせた総合評価では，BART ベースの提案モデルが最高性能を達成した．

表 6：パイプラインの人手評価（5 段階評価の平均値）

	キャラクタ性	流暢性	妥当性	総合評価
NTT	1.12	4.30	3.22	0%
NTT+スタイル変換	3.15	4.14	3.16	5%
rinna+スタイル変換	3.53	4.28	3.59	24%
ChatGPT+スタイル変換	3.62	4.38	4.12	34%
BART+スタイル変換	**3.69**	4.30	3.96	**39%**
ChatGPT（3-shot）	3.46	**4.90**	**4.24**	24%

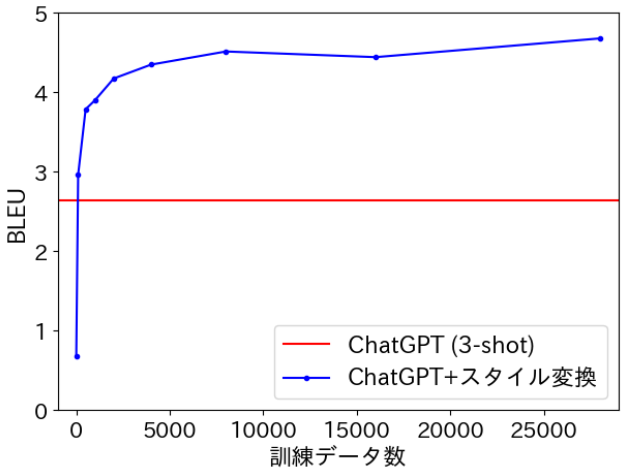

図 1　スタイル変換モデルの訓練に必要な文対数の分析

4.4 分析

4.4.1 スタイル変換モデルの訓練に必要な文対数

スタイル変換モデルの訓練に必要な文対数を明らかにするために，比較的大規模なスタイル変換コーパスを作成できたオカザえもんのキャラクターを対象に，訓練データの規模を縮小しつつパイプライン性能の変化を観察した．具体的には，16,000 件，8,000 件，4,000 件，2,000 件，1,000 件，500 件，100 件，0 件（表 5 のスタイル変換前に対応）と訓練データを減らしつつ，BLEU 値を自動評価した．なお，本実験の応答生成モデルには ChatGPT を用いた．

図 1 に示す実験結果から，8,000 文対ほどの訓練データでスタイル変換の性能は上限に到達し，数百文対ほどの訓練データでさえ大きな効果が得られることが明らかになった．また，比較手法である ChatGPT の few-shot モデルの性能には，100 文対の訓練データのみで到達可能であることも確認できた．これらの分析結果から，数百件から数千件ほど

表 7：BART および ChatGPT による応答生成の例

		スタイル変換前	スタイル変換後
オカザえもん	発話	オカザえもんさんの今週末の予定は?	
	BART	今週末の予定は未定です	今週末の予定は未定でござる
	ChatGPT	今週末は友達と釣りに行く予定です。	今週末は友達と釣りに行く予定でござる
キャベッツさん	発話	こんにちはにゃベ 私も同じもの届きました	
	BART	こんにちはにゃベお仲間ですね	こんにちにゃベ仲間にゃベね
	ChatGPT	あら、それは素敵なご縁ですね。	あら、それはステキな縁にゃベよね
ちぃたん☆	発話	が〜〜〜〜本物のちぃたん☆	
	BART	本物のちぃたん	本物のちぃたん☆ですっ☆
	ChatGPT	ありがとうございます！頑張ります！	ありがとうございますっ！頑張りますっ！
レルヒさん	発話	そのビニールスーツおいくらですか?	
	BART	１万円くらいですかね	1 万円クライダロ？
	ChatGPT	それは 500 円です。	500 円デスカラ。

の発話データさえ入手できれば，大規模な事前訓練を経た応答生成モデルの恩恵を受けつつ，提案手法の枠組みでキャラクタ性を持つ雑談対話を実現できると言える．

4.4.2 定性評価

BART および ChatGPT による応答生成の例を表 7 に示す．特定のキャラクタ性を持たない応答文に対して，スタイル変換によって各キャラクターの特徴が付与されている．これらの事例からも，提案手法の有効性が確認できる．

5. おわりに

本研究では，応答生成とスタイル変換を組み合わせたパイプラインモデルによって，キャラクタ性を持つ雑談対話システムの構築に取り組んだ．ご当地キャラクターの対話データにおける実験の結果，BLEU による自動評価と人手評価において，提案手法の有効性を確認できた．本研究で提案したスタイル変換モデルは効果的に訓練できるため，数百件程度の発話さえあれば，大規模な事前訓練を経た応答生成モデルの恩恵を受けつつ，キャラクタ性を持つ雑談対話を実現できる．

謝辞

データを提供いただいた，岡崎 PR & オカザえもん事業のオカザえもん様，西東京商工会青年部 広報・親善大使のキャベッツさん様，ちぃたん☆様，新潟県エヌキャラネットのレルヒさん様に，深く感謝いたします．

参考文献

[1] Daniel Adiwardana, Minh-Ttang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot. arXiv:2001.09977, 2020.

[2] Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. Recipes for Building an Open-Domain Chatbot. In Proc. of EACL, pp. 300–325, 2021.

[3] Hiroaki Sugiyama, Masahiro Mizukami, Tsunehiro Arimoto, Hiromi Narimatsu, Yuya Chiba, Hideharu Nakajima, and Toyomi Meguro. Empirical Analysis of Training Strategies of Transformer-Based Japanese Chit-Chat Systems. In Proc. of IEEE SLT Workshop, pp. 685–691, 2022.

[4] Reina Akama, Kazuaki Inada, Naoya Inoue, Sosuke Kobayashi, and Kentaro Inui. Generating Stylistically Consistent Dialog Responses with Transfer Learning. In Proc. of IJCNLP, pp. 408–412, 2017.

[5] 柴 淳, 狩野芳伸：キャラクタ的特徴の自動抽出および付与. 人工知能学会第 88 回言語・音声理解と対話処理研究会, pp. 28–33, 2020.

[6] 清水健吾, 上垣貴嗣, 菊池英明：強化学習を用いてキャラクタらしさを付与した雑談応答の生成. 人工知能学会第 94 回言語・音声理解と対話処理研究会, pp. 28–33, 2022.

[7] 宮崎千明, 佐藤理史：発話テキストへのキャラクタ性与のための音変化表現の分類. 自然言語処理, Vol. 26, No. 2, pp. 407–440, 2019.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In Proc. of ICLR, 2022.

[9] Tom B. Brown, njamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. 2020.

[10] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A Survey on Dialogue Systems: Recent Advances and New Frontiers. ACM SIGKDD Explorations Newsletter, Vol. 19, No. 2, pp. 25–35, 2017.

[11] Joseph Weizenbaum. ELIZA: A Computer Program for the Study of Natural Language Communication Between Man and Machine. Communications of the ACM, Vol. 9, No. 1, pp. 36–45, 1966.

[12] Zongcheng Ji, Zhengdong Lu, and Hang Li. An Information Retrieval Approach to Short Text Conversation. arXiv:1408.6988, 2014.

[13] Oriol Vinyals, and Quoc V. Le.: A Neural Conversational Model. In Proc. of ICML, 2015.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In Proc. of NIPS, pp. 6000–6010, 2017.

[15] Alec Radford, Jefferey Wu, Rewon Child, David Luan, Darop Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[16] 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 Proc. of ACL, pp. 7871–7880, 2020.
[17] 杉山弘晃, 成松宏美, 水上雅博, 有本庸浩, 千葉祐弥, 目黒豊美, 中嶋秀治：Transformer encoder-decoder モデルによる趣味雑談システムの構築. 人工知能学会第 90 回言語・音声理解と対話処理研究会, pp. 104–109, 2020.
[18] Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CC-Net: Extracting High Quality Monolingual Datasets from Web Crawl Data. In Proc. of LREC, pp. 4003–4012, 2020.
[19] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models are Zero-Shot Learners. In Proc. of ICLR, 2022.
[20] J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. ParaBank: Monolingual Bitext Generation and Sentential Paraphrasing via Lexically-constrained Neural Machine Translation. In Proc. of AAAI, pp. 6521-6528, 2019.
[21] Tomoyuki Kajiwara, Biwa Miura, and Yuki Arase. Monolingual Transfer Learning via Bilingual Translators for Style-Sensitive Paraphrase Generation. In Proc. of AAAI, pp. 8042–8049, 2020.
[22] Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. Personalized Machine Translation: Preserving Original Author Traits. In Proc. of EACL, pp. 1074–1084, 2017.
[23] Diederik P. Kingma, and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Proc. of ICLR, 2015.
[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In Proc. of ACL, pp. 311–318, 2002.
[25] Katsuhito Sudoh, Kosuke Takahashi, and Satoshi Nakamura. Is This Translation Error Critical?: Classification-Based Human and Automatic Machine Translation Evaluation Focusing on Critical Errors. In Proc. of HumEval, pp.46–55, 2021.

付録

A.1 ChatGPT に入力したプロンプト

```
人間になりきって雑談してください。
以下の入力文に 1 文から 2 文で短めに応答してください。
入力文：（ここに入力文）
応答文：
```

図 A.1　キャラクタ性を考慮しないプロンプト

```
以下の入力例と応答例を参考に、入力文に対して応答してください。
入力例：ゆるキャラグランプリの宣伝、お疲れ様でござる～
応答例：ありがとうでござる～食べたし!
入力例：オカザえもんって 片手立て枕で寝るの??
応答例：イメージ画像でございまする、
入力例：忙しいですか?
応答例：忙しいでござる!
入力文：（ここに入力文）
応答文：
```

図 A.2　オカザえもんの応答例を考慮するプロンプト

```
以下の入力例と応答例を参考に、入力文に対して　応答してください。
入力例：こんにちは 今日も良い 1 日を
応答例：良い 1 日をにゃべー♪
入力例：キャベたん。小さなキャベたんも、いっしょにケーキたべようね
応答例：ケーキ美味しそうにゃべ
入力例：キャベッツさん、2 日間お疲れ様でした
応答例：2 日間、たのしかったにゃべ ありがとにゃべー!!
入力文：（ここに入力文）
応答文：
```

図 A.3　キャベッツさんの応答例を考慮するプロンプト

```
以下の入力例と応答例を参考に、入力文に対して応答してください。
入力例：ちぃたんハマってる時期あったなそういえばwwwww
応答例：またお友達になってくださいっ☆
入力例：格ゲーに興味ありませんか...?
応答例：ストリートファイター大好きですっ☆
入力例：えーんちぃたん☆こわいよぉ
応答例：あやのちゃんっ☆怖くないので今度女子会しましょうねっ☆
入力文：（ここに入力文）
応答文：
```

図 A.4　ちぃたん☆の応答例を考慮するプロンプト

```
以下の入力例と応答例を参考に、入力文に対して応答してください。
入力例：夜中にラーメン、お腹でるでぇ
応答例：デモ 欲望ニハ かなワナイ
入力例：レルヒさん 28 歳だったのか
応答例：ソーソー 永遠ノ 28 歳
入力例：アタマ重いからひっくりカエルぞ!
応答例：重ソーニ見エテ 空ッポヤ
入力文：（ここに入力文）
応答文：
```

図 A.5　レルヒさんの応答例を考慮するプロンプト