Vol.2025-NL-263 No.19
2025/3/9

目標文難易度を連続的に制御可能なテキスト平易化

柳本 大輝[1,a)]　梶原 智之[1,b)]　荒瀬 由紀[2,c)]　二宮 崇[1,d)]

概要：本研究では，言語教育および言語学習の支援を目的に，文難易度の連続的な制御に取り組む．文の難易度制御に関する先行研究は，目標難易度を表現する特殊トークンを入力文頭に加えることによって出力文を変化させており，(1) テキスト平易化モデルは文難易度を暗黙的にしか学習できない，(2) 目標難易度は離散的にしか与えられない，という 2 つの課題を抱えている．より詳細な難易度制御を実現するために，本研究では文ベクトルのノルムとして文難易度を表現することを提案し，これらの課題に対処する．つまり，平易な文ほど小さいノルムを持つようにテキスト平易化モデルの符号化器を訓練しておき，目標難易度に応じて符号化した文ベクトルのノルムを圧縮することで難易度制御を実現する．英語のニュース記事を対象とする評価実験の結果，提案手法は既存手法よりも優れた難易度制御の性能を示した．

1. はじめに

テキスト平易化 [1] では，文の主要な情報を保持したまま，文内の難解な表現を平易な表現に変換することによって，テキストの読みやすさや理解しやすさを改善する．この技術は，言語障害を持つ人々 [2] や子ども [3]，言語学習者 [4] など，多様な人々の文章読解を助けるだけでなく，教育の現場において教師が学生の言語能力を考慮してテキストの難易度を調整する際にも役立つ．ただし，対象読者の言語能力には個人差があり，各読者が求める平易化レベルが異なる点には注意が必要である．特に，子どもや言語学習者の言語能力は年齢や知識量などの影響を受けて様々に変化するが，効率的な言語学習のためには教材テキストの難易度が学習者の言語能力と乖離してはならない [5]．そのため，テキスト平易化において出力文の難易度を制御する手法が近年盛んに研究されている [6–8]．

これらの先行研究 [6–8] では，目標難易度を表現する特殊トークンを入力文頭に加えることによって出力文を変化させている．しかし，これらの既存手法では，文の難易度を明示的に評価する機能をテキスト平易化モデルが内部に備えていない．そのため，これまでのテキスト平易化モデルは，文の難易度を暗黙的にしか学習できていない．また，目標難易度は特殊トークンとして語彙に登録されるため，対象学年などの事前に定義された離散値しか扱えない．

[1] 愛媛大学大学院理工学研究科
[2] 東京科学大学情報理工学院
[a)] yanamoto@ai.cs.ehime-u.ac.jp
[b)] kajiwara@cs.ehime-u.ac.jp
[c)] arase@c.titech.ac.jp
[d)] ninomiya.takashi.mk@ehime-u.ac.jp

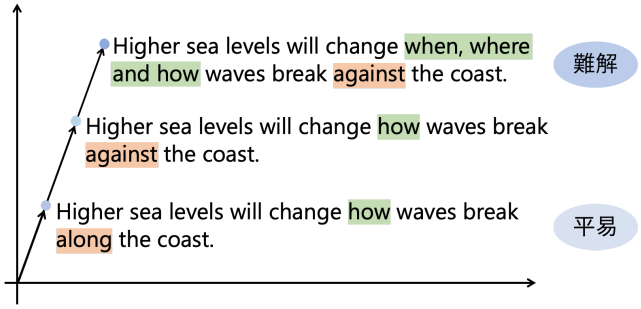

図 1 本研究で扱うベクトル空間のイメージ．文ベクトルは，向きで意味を表現し，長さで難易度を表現する．文ベクトルのノルムが小さいほど平易な文であることを表しており，この例のように同じ向きで長さを変えるとテキスト平易化を実現できる．

テキスト平易化における難易度の制御性を高めるために，本研究では，明示的かつ連続的な文難易度表現機構を備えたテキスト平易化モデルを提案する．提案手法は，Transformer ベースの系列変換モデル [9] であるが，符号化器から文ベクトルを得て，文ベクトルから復号するという構造を持つ．ここで，図 1 に示すように，文ベクトルが向きで意味を表現し，長さで難易度を表現するという特性を獲得するように系列変換モデルを訓練する．このようなベクトル空間が得られれば，符号化器から得た文ベクトルのノルムを目標難易度に応じて圧縮してから復号することによって，出力難易度を柔軟に制御可能なテキスト平易化モデルを構築できる．

提案手法の有効性を検証するために，Newsela-Auto [10] の英語ニュース記事を対象として，難易度推定および難易度制御の 2 つの評価実験に取り組んだ．難易度推定タスクでは，意味的に対応するが難易度の異なる 3 文の組に対

して各文の難易度を推定し，ランキング性能を評価した．難易度制御タスクでは，入力文と目標難易度を与えて難易度を制御し，テキスト平易化の評価指標 SARI [11] を用いて正解文への近似性能を評価した．実験の結果，提案手法は難易度推定の既存手法 [12] および難易度制御の既存手法 [8] をそれぞれ上回り，両タスクにおいて高い性能を達成した．提案モデルは，連続的に難易度を制御できるという明らかな利点を持つだけでなく，難易度推定および難易度制御の性能の面でも既存手法よりも有用である．

2. 関連研究

2.1 文の難易度制御

深層学習に基づくテキスト平易化の先行研究 [13–16] は，機械翻訳と同様の系列変換タスクとして定式化されるのが一般的であり，難解文と平易文の対からなるパラレルコーパス [10,14,17,18] 上での訓練によって，語彙や文法に関する編集操作を獲得する．近年では，入力文中の編集箇所を制御する手法 [19–25] や，文長や構文構造など出力文の形式を制御する手法 [26–29] など，系列変換モデルの制御性の向上によってテキスト平易化の性能が改善されてきた．

言語教育および言語学習の支援に関する応用の目的では，対象読者の言語能力の個人差に対応するために，目標難易度を指定して出力を制御する手法 [6–8] が研究されている．Scarton and Specia [6] は，多言語機械翻訳における出力言語の制御手法 [30] から着想を得て，出力文の目標難易度の制御に初めて取り組んだ．この手法では，入力文頭に目標難易度を示す特殊トークンを追加することによって，同じ入力文に対しても異なる出力が得られる．Nishihara et al. [7] はこのアプローチを改良し，目標難易度の文中で頻出する単語を重み付けし，目標難易度に適した単語の出力を促進することで，難易度制御の性能を改善した．また，Yanamoto et al. [8] は，目標難易度と出力文の推定難易度の誤差を報酬とする強化学習によって，難易度制御の性能を更に改善した．

これらの先行研究では，テキスト平易化モデルは文の難易度をモデル内部で明示的に評価する機能を持たないため，文の難易度は暗黙的にしか捉えられない．Yanamoto et al. [8] の手法においても，報酬関数として出力文の難易度を推定するに留まり，テキスト平易化モデルの内部では難易度を明示的に評価しない．そこで，テキスト平易化モデルの内部に文の難易度を評価する機構を備えられれば，より正確な難易度制御ができると期待できる．また，より詳細な難易度制御のためには，従来の特殊トークンでは表現できない連続的な目標難易度の指定が求められる．

2.2 文の難易度付きコーパス

文の難易度制御に関する先行研究 [6–8] では，英語のニュース記事を対象とする Newsela のパラレルコーパス [10,14,18] が用いられてきた．Newsela パラレルコーパスは，専門家がニュース記事に対して 4 段階の平易化を実施した記事対の中から，自動的な文アライメントの技術によって構築されたものである．各記事には米国学校制度における学年に対応する 2 から 12 までの 11 段階の難易度が付与されているものの，文単位では難易度が付与されていない．先行研究 [6–8] では，文の難易度はその文が含まれる記事の難易度と一致する，と定義してきた．しかし実際には，記事中の各文の難易度は必ずしも一貫しないと考えられるため，より正確な文難易度のラベルを用いて難易度制御を改善できる余地がある．

文単位で難易度が付与されたコーパスとして，CEFR-SP [12] がある．これは，言語運用能力の国際標準である Common European Framework of Reference for Languages (CEFR)[*1] に従って，英文の難易度を 6 段階でアノテーションしたコーパスである．CEFR-SP のアノテータは英語教育の豊富な経験を持つ専門家であるため信頼性の高い文難易度のラベルが付与されているものの，パラレルコーパスではないため，これをテキスト平易化モデルの訓練に直接用いることはできない．

3. 提案手法

本研究では，図 1 に示したように，向きで意味を長さで難易度を表現するベクトル空間を考え，そのベクトル空間上に入力文を写像する符号化器を訓練する．また，ベクトルから文を生成する復号器も同時に訓練する．提案手法は系列変換モデルに基づくが，このような特殊な文ベクトルを構成するため，既存の事前訓練モデル [31,32] の恩恵を受けられない．そこで，3.2 節および 3.3 節で説明する 2 段階の訓練によって，高品質な難易度制御を実現する．

3.1 モデル構成

本研究では，図 2 の左に示すように，符号化器および復号器からなる Transformer ベースの系列変換モデル [9] によって，文の難易度を制御する．ただし，標準的な Transfomer とは異なり，符号化器は文脈化単語ベクトルの系列ではなく文ベクトルを復号器に渡す．この文ベクトルが図 1 のような特性を持つように，次節以降で述べる訓練を行う．

符号化器における文ベクトルの構成には，マスク言語モデル [33] などの先行研究と同様に，文頭に特殊トークン（[CLS]）を付与したり，各トークンに対応するベクトルをプーリングしたりという複数の方法が考えられる．本研究では，[CLS] トークンを文頭に付与し，このトークンに対応する Transformer 符号化器の出力 $v_{[\mathrm{CLS}]}$ を文ベクトルとして用いる．そして，文の難易度を文ベクトルのノルム $\|v_{[\mathrm{CLS}]}\|$ によって表現することとする．

[*1] https://www.coe.int/en/web/common-european-framework-reference-languages

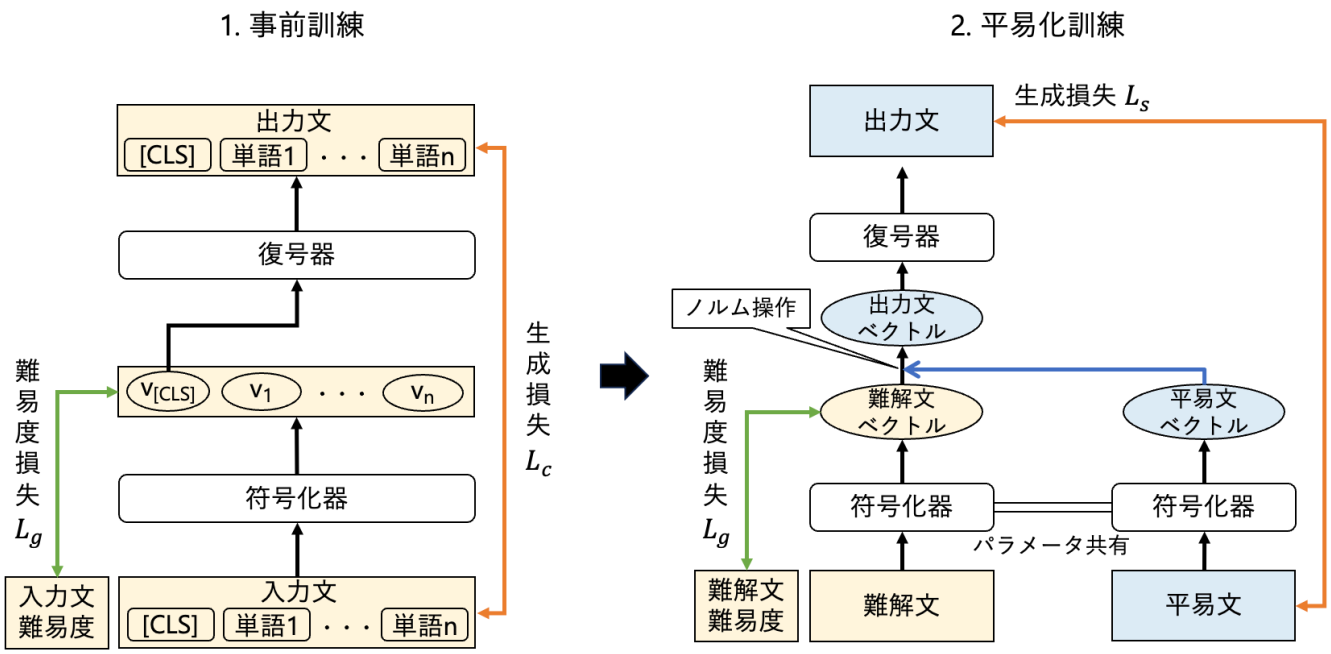

図 2 事前訓練（左）と平易化訓練（右）の 2 段階の訓練からなる提案手法の概要

3.2 事前訓練

提案モデルは特殊な構造を持つため，既存の汎用的な事前訓練モデル [31, 32] を利用できない．しかし，テキスト平易化の訓練に利用可能なパラレルコーパスは多くないため，大規模コーパスを用いた事前訓練によって様々なテキストに触れておくことが重要である．例えば先行研究 [34] では，大規模コーパスを用いた自己符号化の事前訓練によって，テキスト平易化の性能を改善できることが報告されている．そこで本研究でも，提案モデルを用いて自己符号化の事前訓練[*2]を行い，式 (1) によって様々なテキストから文ベクトルへの符号化および文ベクトルからテキストへの復号について学習する．

$$L_c = -\frac{1}{|X|}\sum_{m=1}^{|X|}\log p(x_m|\mathbf{x}_{<m}, v_{[\mathrm{CLS}]}) \tag{1}$$

この自己符号化は，入力文を $X = x_1, x_2, ..., x_{|X|}$ とし，符号化器によって得た文ベクトル $v_{[\mathrm{CLS}]}$ から文 X を復元する際の交差エントロピー損失を最小化することによって訓練する．この損失によって，文ベクトルの向きが意味を表現できるように訓練されることを期待する．

また，本研究では先述のとおり，文ベクトルの長さが難易度を表現できるようにも符号化器を訓練したい．そこで，文 X_i の難易度を g_i として，式 (2) のピアソン相関に基づく損失を最小化する．

$$L_g = 1 - \frac{\sum_{i=1}^{N}(\|v^i_{[\mathrm{CLS}]}\| - \overline{\|v_{[\mathrm{CLS}]}\|})(g_i - \bar{g})}{\sqrt{\sum_{i=1}^{N}(\|v^i_{[\mathrm{CLS}]}\| - \overline{\|v_{[\mathrm{CLS}]}\|})^2}\sqrt{\sum_{i=1}^{N}(g_i - \bar{g})^2}} \tag{2}$$

ここで、N はバッチサイズである．この訓練によって，文難易度と文のノルムの間に正の相関が得られるため，ベクトル空間上で文難易度を表現可能になる．なお，ピアソン相関に基づく損失の代わりに，平均二乗誤差のようなより制約の強い損失を用いることも考えられるが，予備実験によって文の生成に悪影響が見られたため，本研究では難易度を学習するための損失としてピアソン相関を用いる．

最終的に，我々の事前訓練では，式 (1) の生成損失 L_c および式 (2) の難易度損失 L_g の両方を用いるマルチタスク学習を採用し，訓練の優先度を表す重み α を用いて，以下の損失を最小化する．

$$L_p = \alpha L_c + L_g \tag{3}$$

この事前訓練には，テキスト平易化のためのパラレルコーパスは必要なく，文およびその難易度ラベルのみを用いる．

3.3 平易化訓練

前節の方法で事前訓練した系列変換モデルを，テキスト平易化のパラレルコーパス上で追加訓練し，難易度制御モデルを得る．図 2 の右に示すように，まず，テキスト平易化パラレルコーパスから得た難解文 X および平易文 Y の文対を用いて，難解文のベクトル $v^X_{[\mathrm{CLS}]}$ のノルムが平易文のベクトル $v^Y_{[\mathrm{CLS}]}$ のノルムと一致するように，式 (4) のノルム操作を行う．

[*2] 先行研究 [34] では，自己符号化よりも折り返し翻訳を復元する事前訓練の方が有用だと報告されているが，本研究ではより低コストに実施できる自己符号化の事前訓練を採用する．

表 1　データ件数

	訓練	検証	評価
C4	10,000,000 文	10,000 文	-
Newsela-Auto	394,300 文対	43,317 文対	44,067 文対

$$v'_{[CLS]} = v^X_{[CLS]} * \frac{\|v^Y_{[CLS]}\|}{\|v^X_{[CLS]}\|} \tag{4}$$

この文ベクトル $v'_{[CLS]}$ は，向きは難解文の意味を表現しつつ，長さは平易文の難易度を表現する．そして，この文ベクトルから平易文を生成するために，以下の交差エントロピー損失を最小化する．

$$L_s = -\frac{1}{|Y|}\sum_{m=1}^{|Y|} \log p(y_m|\mathbf{y}_{<m}, v'_{[CLS]}) \tag{5}$$

この訓練によって，ノルム操作によって目標難易度を指定した上でテキスト平易化を行う難易度制御モデルを獲得できる．前節と同様に，この平易化訓練においても，式 (5) の生成損失 L_s および式 (2) の難易度損失 L_g の両方を用いるマルチタスク学習を採用し，訓練の優先度を表す重み β を用いて，以下の損失を最小化する．

$$L_f = \beta L_s + L_g \tag{6}$$

4. 評価実験

提案モデルについて，英語の難易度推定および難易度制御の両タスクにおける性能を評価する．

4.1 データセット

事前訓練のための大規模な英語コーパスとして，Colossal Clean Crawled Corpus (C4) [32] を使用した．C4 は，系列変換モデル T5 [32] の事前訓練のために Common Crawl からスクレイピングされたテキストで構築されている．様々なフィルタリングが適用されており，重複している文や不適切な表現を含む文などが取り除かれている．本コーパスに含まれる各文に対して，訓練済みの文難易度推定器[*3] [12] を用いて，6 段階の CEFR 難易度を付与して事前訓練に使用した．本実験では，CEFR 難易度ごとに文数が均等になるように，訓練用に 1 千万文および検証用に 1 万文を抽出して利用した．

平易化訓練のためのテキスト平易化パラレルコーパスとして，英語のニュース記事を対象とする Newsela-Auto [10] を公式の分割に従って使用した．本コーパスも，各文に対して 6 段階の CEFR 難易度を付与して使用した．データ件数を表 1 に示す．

4.2 評価方法

難易度ランキングタスクでは，モデルが難易度ごとに文を正しく並び替えられるかどうかを評価した．まず，Newsela データセット内で同じタイトルを持つ，平易度の異なる 5 パターンの記事から，それぞれ 1 文ずつ，同じ意味を持つ文を抽出して 5 文セットを作成した．隣接する平易度の記事から得られた文同士は難易度の差が小さいことが考えられるため，5 文セットのうち，平易化されていない元記事から得られた文，2 段階平易化された記事から得られた文，4 段階平易化された記事から得られた文の 3 文を抽出した．この 3 文に対して難易度推定を行い，推定難易度が高い順に並び替え，その順番が記事の平易度の順番と一致するかどうかを評価した．評価指標としては，nDCG，スピアマンの順位相関を使用した．

表 2　文難易度ランキングにおける評価結果

モデル	nDCG	Spearman r
既存手法	0.965	0.868
回帰モデル	0.981	0.822
提案手法	0.972	0.772
提案手法（事前訓練のみ）	0.859	-0.397
提案手法（平易化訓練のみ）	**0.986**	**0.882**

テキスト平易化タスクでは，Newsela-Auto の評価セットに対する平易化モデルの出力を評価した．評価指標には，このタスクで広く使用されている評価指標である SARI を使用し，自動評価パッケージ EASSE [35] を用いて評価する．SARI は，単語 n-gram の追加 (add)・保持 (keep)・削除 (del) 操作の F 値の平均値であり，詳細な分析のために平均する前の各評価値も確認する．

4.3 実装

提案手法の実装には，pytorch，pytorch-lightning，huggingface transformers[*4] [36] を用いた．提案手法モデルはエンコーダデコーダモデルであり，エンコーダとデコーダのどちらも Transformer ベースの構造を持つ．エンコーダおよびデコーダのレイヤ数は 12，注意機構のヘッド数は 12，埋め込み次元数は 768，全結合層の次元数は 3072 とし，Dropout 率は 0.1 とした．エンコーダのパラメータのみ，事前訓練モデルである bert-base-cased [33][*5]のパラメータを用いて初期化した．また，エンコーダとデコーダの埋め込み層，Softmax 層直前の線形変換層の 3 つのパラメータを共有した．

事前訓練

バッチサイズを 128，学習率を 1×10^{-4}，最適化手法には AdamW [37] を使用し，warmup ステップは 10,000 に設定した．また，検証用データにおける損失値が 5 エポック連続で低下しない場合に訓練を終了する early stopping を適用した．難易度損失と生成損失の重みは，$\alpha = 1$ とした．

[*3] https://github.com/yukiar/CEFR-SP
[*4] https://github.com/huggingface/transformers
[*5] https://huggingface.co/google-bert/bert-base-cased

表 3　テキスト平易化タスクにおける評価結果

	SARI	add	keep	del
入力文そのまま	12.04	0.00	36.12	0.00
Transformer + 目標難易度の指定	41.10	3.35	42.90	77.04
Transformer + 目標難易度の指定 + 単語難易度の考慮	41.50	**3.44**	42.97	78.07
Transformer + 目標難易度の指定 + 強化学習	41.96	3.41	42.22	80.24
提案手法	**43.34**	3.06	**45.13**	81.82
提案手法（事前訓練のみ）	17.80	0.24	36.08	17.06
提案手法（平易化訓練のみ）	33.33	1.81	11.12	**87.05**

平易化訓練

バッチサイズを 64，学習率を 1×10^{-5}，最適化手法には AdamW [37] を使用し，warmup ステップは 4,000 に設定した．また，検証用データにおける SARI スコアが 10 回連続で低下しない場合に訓練を終了する early stopping を適用した．検証は 1/4 エポックごとに実行した．難易度損失と生成損失の重みは，$\beta = 1$ に設定した．

4.4　比較手法

難易度ランキングタスクでは，訓練済みの文難易度推定器[*3] [12] を比較手法として用いた．また，文難易度と予測値の平均二乗誤差を損失関数とするシンプルな BERT 回帰モデルを訓練した．この回帰モデルの訓練データセットは，Newsela-Auto に含まれる難解文と平易文の対から文を重複なく抽出して作成した．

テキスト平易化タスクにおける実験では，提案手法と既存手法である文頭に目標難易度トークンを追加する手法の性能を比較した．Newsela の難易度を用いて訓練したモデルとして，Scarton ら [6] の手法を用いたモデル (Transformer+grade)，西原ら [7] の手法を用いたモデル (Transformer+grade+word)，柳本ら [8] の手法を用いたモデル (Transformer+grade+reinforce) の 3 つと比較する．また，CEFR の難易度を用いた場合の Scarton らの手法とも比較する．さらに，提案手法のバリエーションとして，事前訓練のみを行った場合と平易化訓練のみを行った場合の 2 つのモデルとも比較する．

5.　実験結果

5.1　難易度ランキングタスク

難易度ランキングタスクにおける評価結果を表 2 に示す．既存手法は，nDCG およびスピアマンの順位相関の両方で高い性能を示した．また，文難易度の予測に特化したシンプルな回帰モデルも高い性能を示した．提案手法では，事前訓練のみのモデルが最も低い性能となった一方で，平易化訓練のみのモデルが最も高い性能を示した．2 段階の訓練を経たモデルは，既存手法を nDCG は上回ったが，スピアマンの順位相関位おいては 0.1 ポイントほど下回った．既存手法は分類モデルであり，異なる文に対して同じ難易度を出力する傾向が強かった．その結果，順位を明確に評価するランキング指標では性能が低下するものの，難易度の順位が逆転することはなく，順位相関への影響は限定的であった．しかし，難易度制御においては細かな難易度の差を捉えることが重要であり，nDCG のような明確な並び替えの正確さを評価する指標のスコアが高くなることが望ましい．

5.2　テキスト平易化タスク

テキスト平易化タスクでの評価結果を表 3 に示す．2 段階の訓練を経た提案手法モデルが，最も高い SARI スコアを示した．また，SARI における add，keep，del の評価値を見ると，既存手法よりも keep と del は 1 ポイント以上上回る一方で，add は 0.4 ポイント程度下回った．

事前訓練のみの場合，SARI は 17.80 にとどまった．各操作の F 値をみても，keep が入力文の値と非常に近く，del が著しく低いことから，事前訓練段階では，ノルムを変化させても出力文は入力文からほとんど変化せず，モデルが保守的なことがわかった．これは，事前訓練では入力文の再構成を目的として訓練しており，ノルムの変化と平易化操作の関係を明示的に訓練していないことが原因の 1 つとして考えられる．平易化訓練のみの場合，add や keep が他のモデルを大きく下回る結果となった．単一の文ベクトルから文を生成させる場合，事前訓練が有効であることが示唆される．

6.　おわりに

本研究では，文難易度を連続的に制御可能なテキスト平易化モデルを提案した．提案モデルは，入力文に応じた文ベクトルをモデル内部で明示的に生成し，文ベクトルのノルムを用いて文難易度を表現する．さらに，文ベクトルに基づいて出力文を構成する．提案手法では，事前訓練および平易化訓練の 2 段階訓練を通じて，モデルをテキスト平易化タスクに適応させる．

英語テキスト平易化のためのパラレルコーパス Newsela-Auto を用いて，文難易度ランキングタスクとテキスト平易化タスクの 2 つのタスクで提案手法の有効性を検証した．両タスクの評価実験を通じて，提案手法は複数の文におけ

表 4 意味的に対応するが難易度の異なる例

CEFR レベル	文	文ベクトルのノルム
B2	Sven Spannekrebs, their coach, says the sisters are making amazing progress, though he is realistic about their prospects as athletes.	1.62
B1	Sven Spannekrebs, their coach, says the sisters are making amazing progress.	1.00
A2	Their coach says the sisters are doing very well.	0.60

表 5 ノルムが難解文と平易文の中間に位置するベクトルからの生成例

		文ベクトルのノルム
難解文	In Nepal, for example, thousands of female health volunteers visit homes to provide immunizations, family planning materials and information on infant care.	1.38
中間文	In Nepal, for example, thousands of female health volunteers visit homes to provide vaccines and information on baby care.	0.90
平易文	In Nepal, for example, thousands of female health volunteers visit homes.	0.70

る細かな難易度の差を捉え，高精度で並び替えを実行でき，既存手法よりも直感的な出力の難易度制御が可能であることが実証された．

謝辞 本研究は JSPS 科研費（基盤研究 B，課題番号：JP23K21732）の助成を受けたものです．

参考文献

[1] Alva-Manchego, F., Scarton, C. and Specia, L.: Data-Driven Sentence Simplification: Survey and Benchmark, *Computational Linguistics*, Vol. 46, No. 1, pp. 135–187 (2020).

[2] Carroll, J., Minnen, G., Canning, Y., Devlin, S. and Tait, J.: Practical Simplification of English Newspaper Text to Assist Aphasic Readers, *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pp. 7–10 (1998).

[3] De Belder, J. and Moens, M.-F.: Text Simplification for Children, *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pp. 19–26 (2010).

[4] Petersen, S. E. and Ostendorf, M.: Text Simplification for Language Learners: A Corpus Analysis, *Proceedings of the Workshop on Speech and Language Technology in Education*, pp. 69–72 (2007).

[5] Laufer, B.: How Much Lexis is Necessary for Reading Comprehension?, *Vocabulary and Applied Linguistics*, pp. 126–132 (1992).

[6] Scarton, C. and Specia, L.: Learning Simplifications for Specific Target Audiences, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 712–718 (2018).

[7] Nishihara, D., Kajiwara, T. and Arase, Y.: Controllable Text Simplification with Lexical Constraint Loss, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 260–266 (2019).

[8] Yanamoto, D., Ikawa, T., Kajiwara, T., Ninomiya, T., Uchida, S. and Arase, Y.: Controllable Text Simplification with Deep Reinforcement Learning, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pp. 398–404 (2022).

[9] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is All you Need, *Proceedings of the 31st Conference on Neural Information Processing Systems*, pp. 5998–6008 (2017).

[10] Jiang, C., Maddela, M., Lan, W., Zhong, Y. and Xu, W.: Neural CRF Model for Sentence Alignment in Text Simplification, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7943–7960 (2020).

[11] 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).

[12] Arase, Y., Uchida, S. and Kajiwara, T.: CEFR-Based Sentence Difficulty Annotation and Assessment, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6206–6219 (2022).

[13] 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).

[14] Zhang, X. and Lapata, M.: Sentence Simplification with Deep Reinforcement Learning, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 584–594 (2017).

[15] Zhao, S., Meng, R., He, D., Saptono, A. and Parmanto, B.: Integrating Transformer and Paraphrase Rules for Sentence Simplification, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3164–3173 (2018).

[16] Maddela, M., Alva-Manchego, F. and Xu, W.: Controllable Text Simplification with Explicit Paraphrasing, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3536–3553 (2021).

[17] 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).

[18] 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).

[19] Kajiwara, T.: Negative Lexically Constrained Decoding for Paraphrase Generation, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6047–6052 (2019).
[20] Dong, Y., Li, Z., Rezagholizadeh, M. and Cheung, J. C. K.: EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3393–3402 (2019).
[21] Mallinson, J., Severyn, A., Malmi, E. and Garrido, G.: FELIX: Flexible Text Editing Through Tagging and Insertion, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1244–1255 (2020).
[22] Omelianchuk, K., Raheja, V. and Skurzhanskyi, O.: Text Simplification by Tagging, *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 11–25 (2021).
[23] Agrawal, S., Xu, W. and Carpuat, M.: A Non-Autoregressive Edit-Based Approach to Controllable Text Simplification, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3757–3769 (2021).
[24] Zetsu, T., Kajiwara, T. and Arase, Y.: Lexically Constrained Decoding with Edit Operation Prediction for Controllable Text Simplification, *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability*, pp. 147–153 (2022).
[25] Zetsu, T., Arase, Y. and Kajiwara, T.: Edit-Constrained Decoding for Sentence Simplification, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7161–7173 (2024).
[26] Martin, L., de la Clergerie, É., Sagot, B. and Bordes, A.: Controllable Sentence Simplification, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4689–4698 (2020).
[27] Sheang, K. C. and Saggion, H.: Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer, *Proceedings of the 14th International Conference on Natural Language Generation*, Association for Computational Linguistics, pp. 341–352 (2021).
[28] Martin, L., Fan, A., de la Clergerie, É., Bordes, A. and Sagot, B.: MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 1651–1664 (2022).
[29] Agrawal, S. and Carpuat, M.: Controlling Pre-trained Language Models for Grade-Specific Text Simplification, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12807–12819 (2023).
[30] 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).
[31] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880 (2020).
[32] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P. J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67 (2020).
[33] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186 (2019).
[34] Kajiwara, T., Miura, B. and Arase, Y.: Monolingual Transfer Learning via Bilingual Translators for Style-Sensitive Paraphrase Generation, *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pp. 8042–8049 (2020).
[35] Alva-Manchego, F., Martin, L., Scarton, C. and Specia, L.: EASSE: Easier Automatic Sentence Simplification Evaluation, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 49–54 (2019).
[36] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q. and Rush, A.: Transformers: State-of-the-Art Natural Language Processing, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45 (2020).
[37] Loshchilov, I. and Hutter, F.: Decoupled Weight Decay Regularization, *Proceedings of the Seventh International Conference on Learning Representations* (2019).