Vol.2022-NL-253 No.17
2022/9/30

日本語文法誤り訂正評価コーパスへの誤用タグ付け

小山 碧海[1,a)]　喜友名 朝視顕[1]　三田 雅人[2,1]　岡 照晃[1]　小町 守[1,b)]

概要：本稿では日本語文法誤り訂正評価コーパスへ誤用タグ付けを行う．日本語学習者が犯す誤りには助詞誤りや時制誤りなど様々な種類の誤りがある．対して日本語文法誤り訂正評価コーパス TMU Evaluation Corpus for Japanese Learners (TEC-JL) には誤りの種類を詳細にする誤用タグが付与されていない．文法誤り訂正モデルには訂正しやすい誤りとしにくい誤りが存在し，モデルの誤りタイプ別の評価に誤用タグは必要不可欠である．そこで本研究では TEC-JL 中の各誤りに誤用タグを付与し，モデルを誤りタイプ別に評価可能にする．そして付与した誤用タグを利用し，日本語文法誤り訂正モデルを誤りタイプ別に評価した結果を報告する．

1. はじめに

文法誤り訂正とは，与えられた文章中の文法誤りを文法的に正しい表現に訂正するタスクである[*1]．文法誤り訂正では近年，機械翻訳に基づく手法が盛んに研究されている [1], [2], [3], [4], [5], [6], [7]．以前のルール [8] や分類器 [9] に基づく手法では訂正する誤りを限定していたが，機械翻訳に基づく手法では誤りを限定せずに訂正可能になった．しかし機械翻訳に基づく手法にはモデルによって訂正しやすい誤りとしにくい誤りが存在する．例えば英語文法誤り訂正では，CNN [10] に基づくモデルは局所的な情報で訂正可能な誤りを訂正しやすい傾向がある [2]．また Transformer [11] に基づくモデルは CoNLL-2014 shared task [12] の評価コーパス上で時制誤りの訂正性能が高いことが報告されている [13]．さらに BEA-2019 shared task [14] では，機械翻訳に基づく手法の傾向として綴り誤りや主語と動詞の一致誤りが訂正しやすい誤りタイプだと明らかになった．一方，名詞や形容詞などの語彙選択誤りは訂正しにくい誤りタイプであることも判明した．こうした誤りタイプ別の評価を行うには，評価コーパスに誤りを分類する誤用タグが付与されている必要がある．

そこで本稿では日本語文法誤り訂正評価コーパス TMU Evaluation Corpus for Japanese Learners (TEC-JL) [15] へ行った誤用タグ付けについて述べる．現在，TEC-JL には誤用タグが付与されておらず，モデルを誤りタイプ別に評価することができない．我々は TEC-JL 中の各誤りに誤用タグを付与し，日本語文法誤り訂正モデルを誤りタイプ別に評価可能にした．付与した誤用タグを利用し，モデルを誤りタイプ別に評価した結果を報告する．

本研究の主な貢献は以下の通りである．

- 日本語文法誤り訂正評価コーパス TEC-JL に誤用タグを付与した．
- 日本語文法誤り訂正モデルを誤りタイプ別に評価した結果を報告した．

2. 関連研究

日本語学習者作文に誤用タグを付与したコーパスに NAIST 誤用コーパス [16] がある．NAIST 誤用コーパスは作文対訳データベース [17] という日本語学習者コーパスに誤用タグを付与したコーパスである．NAIST 誤用コーパスでは 76 種類の誤用タグを設計している．具体的には助詞や動詞といった品詞に基づく誤用タグやコロケーションや文体[*2]といった誤用タグなどがある．なたね[*3]でも日本語学習者作文に誤用タグを付与している．なたねでは 77 種類の誤用タグを設計しており，助詞や動詞といった品詞に基づく誤用タグや誤用の要因・背景に関する誤用タグがある．NAIST 誤用コーパスやなたねの誤用タグは学習者の誤用分析や学習者へのフィードバックを目的に設計されており，文法誤り訂正モデルの分析には粒度が細かい．そこで我々は日本語文法誤り訂正モデルの分析を目的とした誤用タグを新たに設計し TEC-JL 中の各誤りに付与した．

[1] 東京都立大学
1–1 Minamiosawa, Hachioji, Tokyo 192–0397, Japan
[2] 株式会社サイバーエージェント
Abema Towers 40–1 Udagawacho, Shibuya, Tokyo 150–0042, Japan
[a)] koyama-aomi@ed.tmu.ac.jp
[b)] komachi@tmu.ac.jp

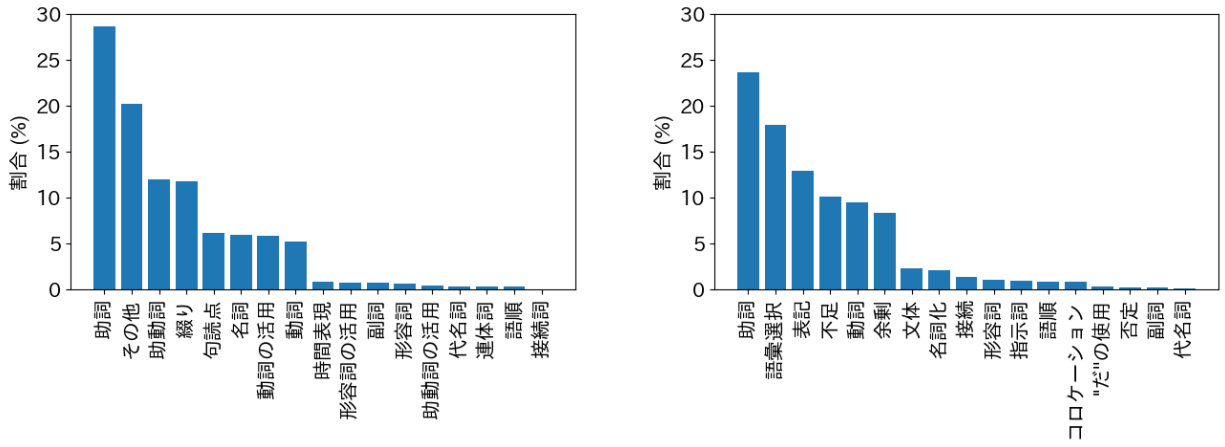
[*1] 本稿では，便宜上，文法誤りだけでなく綴りや内容語の誤りも訂正対象に含める．

[*2] 常体（だ・である体）と敬体（です・ます体）の一致誤り．

[*3] https://hinoki-project.org/natane

表 1: 本研究で設計した誤用タグの一覧.

行	誤用タグ	意味	例
1	ADJ	形容詞の誤り	そんなことは一番 { **大切な** → **重大な** } 欠点だと思うでしょう。
2	ADV	副詞の誤り	そして、自分の開発とテストの仕事を { **よく** → **しっかり** } 完遂しました。
3	AUX	助動詞の誤り	私は日本語が大好き { **ます** → **です** }。
4	CONJ	接続詞の誤り	{ **しかし** → **そして** }、私の一番の夢はタイへ行って、タイ語で現地の人と交流することです。
5	DET	指示詞の誤り	{ **あの** → **その** } 化学会社の名前はチッソ株式会社だった。
6	NOUN	名詞の誤り	その人はソフトウェアの { **権力** → **権利** } を持っている。
7	PART	助詞の誤り	今日、雷の音 { **に** → **で** } 起きました。
8	PRON	代名詞の誤り	まぁ、今回の話は { **そこ** → **ここ** } までにします。
9	PUNCT	句読点の誤り	一緒に歌を歌いました {、→。} ご飯を食べて、買い物をしました。
10	VERB	動詞の誤り	朝御飯を { **飲み** → **食べ** } ました。
11	ADJ:INFL	形容詞の活用の誤り	突然とても { **寂しい** → **寂しく** } 感じ始めました。
12	AUX:INFL	助動詞の活用の誤り	しゅくだいはとてもつまら { **ない** → **なく** } てとてもむずかしいです。
13	VERB:INFL	動詞の活用の誤り	享受 { **する** → **し** } ながら、進める。
14	VERB:TENSE	時間表現の誤り	知っ { **た** → **ている** } 人気ウェブがあるなら、ぜひお知らせください。
15	SPELL	綴りの誤り	国内の { **メデイア** → **メディア** } も管理されて過激な言論はいっさい禁止されています。
16	WO	語順の誤り	日本語 { **四級検定** → **検定四級** } 合格!
17	OTHER	その他の誤り	数秒後、ラベンダーは { **見えるような速度を立て** → **見る見るうちに立って** } いきます。



(a) TEC-JL **(b)** NAIST 誤用コーパス

図 1: 各コーパス中の誤りタイプの割合.

3. 誤用タグの付与

我々は日本語文法誤り訂正モデルの分析を目的に誤用タグを新たに設計し TEC-JL 中の各誤りに付与した．誤用タグは英語文法誤り訂正で代表的な誤用タグ付けツール ERRANT [18], [19] の誤用タグを参考に，Universal Dependencies (UD) [20] の日本語コーパスの品詞体系 [21] に基づき設計した．UD は言語横断的に品詞や構文構造をアノテーションするための枠組みである [22], [23]．UD の品詞体系に基づくことで，日本語にあまり馴染みのない研究者が本評価コーパスを利用する場合でも誤用タグの意味を理解しやすくなる．多言語にわたって文法誤り訂正の研究を行う場合でも，言語間で誤用タグの設計が共通していればモデルの誤りタイプ別の分析が容易になる．

表 1 に本研究で設計した誤用タグの一覧を示す．1 行目から 10 行目は語彙選択に関する誤用タグである．語彙選択の誤用タグはある品詞の単語が同じ品詞の別の単語に訂正された場合に付与される．例えば 1 行目では，“大切な”

表 2: 綴り誤りの例.

翻字の誤り	
学習者文	明日大学の後に「**シェルロク・ホールムス**」をみます。
訂正文	明日大学の後に「**シャーロック・ホームズ**」をみます。
仮名漢字変換システムに起因する誤り	
学習者文	**張る**のが暖かいですから。
訂正文	**春**のほうが暖かいですから。

表 3: 実験で使用したデータセットの詳細.

データセット	原文数	参照文数	用途	評価器
Lang-8 コーパス	1,042,932	1	訓練	-
FLUTEC-dev	1,047	5	開発	-
本研究の評価コーパス (TEC-JL)	1,702	2	評価	M^2 scorer
FLUTEC-test	1,029	5	評価	$GLEU^+$
NAIST 誤用コーパス	6,587	1	評価	M^2 scorer
Wiki-40B	10,000,000	-	-	-

という形容詞が “重大な” という別の形容詞に訂正されているため ADJ の誤用タグが付与される[*4]．11 行目から 13 行目は活用に関する誤用タグである．活用の誤用タグはある単語が別の活用形に訂正された場合に付与される．例えば 11 行目では，“寂しい” という形容詞が “寂しく” という連用形に訂正されているため ADJ:INFL の誤用タグが付与される．14 行目の VERB:TENSE の誤用タグはテンスやアスペクトといった時間表現について訂正された場合に付与される．15 行目の SPELL の誤用タグは誤字・脱字・衍字などの誤りが訂正された場合に付与される．また SPELL には翻字の誤りも含める．16 行目の WO の誤用タグは隣り合う単語を入れ替えるような訂正がされた場合に付与される．17 行目の OTHER の誤用タグは 1 行目から 16 行目の誤用タグの中に該当する誤用タグがなかった場合に付与される．

4. NAIST 誤用コーパスとの比較

TEC-JL と NAIST 誤用コーパス中の各誤りタイプの割合を図 1 に示す[*5]．TEC-JL と NAIST 誤用コーパスでは誤りタイプの粒度が異なるため厳密な比較は困難であることに注意する必要がある．比較の結果，どちらも助詞誤りの割合が最も大きい．日本語学習者にとって助詞を使いこなすことが難しいと分かる．また TEC-JL 中の綴り誤り（NAIST 誤用コーパスでの表記誤りに相当）の割合が NAIST 誤用コーパスと同様に比較的高い．綴り誤りには翻字の誤りや仮名漢字変換システムに起因する誤りが見られた．表 2 に翻字の誤りと仮名漢字変換システムに起因する誤りの例を示す．翻字の誤りでは，学習者が “シェルロク・ホールムス” と書いたのを “シャーロック・ホームズ” に訂正している．翻字は目的言語での表記をあらかじめ知っておかないと正しく表記することが難しく，学習者にとって間違いやすい誤りである．仮名漢字変換システムに起因する誤りでは，学習者の誤入力 “張る” を “春” に訂正している．NAIST 誤用コーパスは手書き作文を基にしているため誤入力は起きず，Lang-8 コーパスを基にしている TEC-JL 特有の誤りである．

5. 実験

本節では代表的な文法誤り訂正モデルを使用し，日本語文法誤り訂正モデルの訂正傾向を調査する．

5.1 実験設定

5.1.1 データセット

表 3 に本実験で使用したデータセットを示す．訓練データには Lang-8 コーパス [29] を使用した．ただし TEC-JL と FLUTEC [30] に使用されている文章は取り除いた[*6]．

[*4] 本研究では ERRANT と同様に自然言語処理ライブラリ spaCy [24] が出力した品詞タグに基づき誤用タグを付与した．近年 ERRANT を基にした誤用タグ付けツールがドイツ語 [25] やルーマニア語 [26]，ギリシャ語 [27]，チェコ語 [28] など様々な言語で開発されている．日本語でも同様の誤用タグ付けツールを開発する場合，性能評価に本評価コーパスを利用可能にするため ERRANT に合わせ spaCy を使用した．

[*5] NAIST 誤用コーパスは大山ら [16] が実験で用いた誤りタイプのみを記載した．

[*6] TEC-JL と FLUTEC は Lang-8 コーパスから作成した評価コーパスであり，リークを防ぐために取り除いた．

表 4: 各文法誤り訂正モデルの性能.

	TEC-JL			FLUTEC	NAIST 誤用コーパス		
モデル	Prec.(%)	Rec.(%)	$F_{0.5}$	$GLEU^+$	Prec.(%)	Rec.(%)	$F_{0.5}$
訂正なし	N/A	0.00	N/A	51.42	N/A	0.00	N/A
CNN	**53.57**	23.22	**42.46**	53.35	**35.16**	4.78	15.47
Transformer	39.92	**39.13**	39.75	**55.39**	30.97	**9.23**	**21.05**

水本ら [29] と同様に，文長制限などを施し，訓練時にノイズとなる文対を除去した．開発データには FLUTEC の開発データを用いた．水本 [31] の実験結果に基づき，学習者文は文字に分割し，訂正文は語彙サイズ 16,000 で 1-gram 言語モデルに基づくトークン化 [32] を施した．1-gram 言語モデルに基づくトークン化を行う際の実装には SentencePiece [33] を使用した[*7].

5.1.2 性能評価

書き手の習熟度や作文のトピックなどによって多様な入力が想定される文法誤り訂正では，複数の評価コーパスでモデルを多面的に評価することが望ましい [13]．そこで評価データには TEC-JL と FLUTEC，NAIST 誤用コーパスを使用した．TEC-JL と FLUTEC は元データに Lang-8 コーパスを用いている点で共通している．しかし TEC-JL は学習者文へ文法的に正しくなる最小限の訂正を施している一方で，FLUTEC は文法的に正しいだけでなく流暢になる訂正 [34] を施しているという違いがある．そのため TEC-JL ではモデルの文法性に絞った評価ができ，FLUTEC では文法性に加え流暢性を評価できる．また NAIST 誤用コーパスは Lang-8 コーパスとドメインが異なるため[*8]，訓練データと評価データのドメインの違いによる性能差を評価できる．TEC-JL と NAIST 誤用コーパスは Max Match (M^2) [35] で評価し，FLUTEC は $GLEU^+$ [36] で評価した[*9]．単語分割誤りが評価結果に影響を与えないようにするため文字単位で評価した．報告する全ての値は 4 つの異なるシードで訓練されたモデルのスコアの平均である．各評価尺度の説明は以下の通りである．

M^2 [35] M^2 は文法誤り訂正で最も一般的な評価尺度である．M^2 ではモデルの出力文が行った編集を参照文の編集となるべく多く一致するように計算し，Precision，Recall，$F_{0.5}$ を求める[*10]．M^2 の計算には M^2 scorer を使用した[*11].

$GLEU^+$ [36] $GLEU^+$ は機械翻訳において代表的な評価尺度 BLEU [37] を文法誤り訂正用に改良した評価尺度である．BLEU がモデルの出力文と参照文の N-gram のみを使用するのに対し，$GLEU^+$ では原文の N-gram も使用しスコアリングする．$GLEU^+$ の計算には Napoles ら [36] が公開しているスクリプトを使用した[*12].

5.1.3 文法誤り訂正モデル

英語文法誤り訂正では，CNN に基づくモデル [2], [38] や Transformer に基づくモデル [4], [39] が代表的である．そのため本実験では CNN と Transformer を文法誤り訂正モデルに使用した．各モデルの実装には fairseq[*13] [40] を利用した．各モデルの設定は以下の通りである．

CNN [10] アーキテクチャは Gehring ら [10] のモデルをベースにした 6 層のエンコーダ・デコーダモデルである．エンコーダとデコーダの単語埋め込みは 512 次元であり，カーネルサイズは 3 である．訓練時の最適化方法や推論時の設定は Kiyono ら [41] に従った．

Transformer [11] アーキテクチャは Vaswani ら [11] のモデルをベースにした 6 層のエンコーダ・デコーダモデルである．具体的には “Transformer (base)” と同様であり，エンコーダとデコーダの単語埋め込みは 512 次元である．訓練時の最適化方法や推論時の設定は Kiyono ら [41] に従った．

5.2 実験結果

表 4 に各文法誤り訂正モデルの性能を示す．訂正なしは入力文をモデル出力とみなし評価した時のスコアである．表 4 より，CNN は Precision が高く，不必要な訂正が Transformer よりも少ないことが分かる．その結果，最小限の訂正を施した TEC-JL では CNN の $F_{0.5}$ が最も高くなった．一方 Transformer は Recall が高いことが分かる．また流暢な訂正を施した FLUTEC でも，CNN より Transformer の方が $GLEU^+$ スコアが高い．したがって Transformer は流暢な訂正を行っていると考えられる．NAIST 誤用コーパスでは TEC-JL と比べ，各モデルのスコアが低いことが分かる．訓練データである Lang-8 コー

[*7] https://github.com/google/sentencepiece
[*8] NAIST 誤用コーパスの作文は課題作文である一方，Lang-8 コーパスの作文は自由作文である点などが異なる．
[*9] 英語文法誤り訂正では，最小限の訂正を施した CoNLL-2014 shared task の評価コーパスは M^2 で評価し，流暢な訂正を施した JFLEG は $GLEU^+$ で評価していることを考慮した．
[*10] 文法誤り訂正では CoNLL-2014 shared task 以降，Precision を重視した $F_{0.5}$ を使用することが一般的である．
[*11] https://github.com/nusnlp/m2scorer
[*12] https://github.com/cnap/gec-ranking
[*13] https://github.com/facebookresearch/fairseq

表 5: TEC-JL における各モデルの誤りタイプ別の Recall (%).

	誤りタイプ							
モデル	助詞	助動詞	綴り	句読点	名詞	動詞の活用	動詞	その他
CNN	29.34	21.91	42.79	6.57	7.80	24.30	19.45	10.62
Transformer	49.12	36.95	49.60	38.19	18.34	37.82	38.67	24.53

表 6: 擬似データを用いた場合の各モデルの性能．**太字**は各カラムでの最高性能を表し，下線は各モデルでの最高性能を表す．

		TEC-JL			FLUTEC	NAIST 誤用コーパス		
行	モデル	Prec.(%)	Rec.(%)	$F_{0.5}$	$GLEU^{+}$	Prec.(%)	Rec.(%)	$F_{0.5}$
1	CNN	53.57	23.22	42.46	53.35	35.16	4.78	15.47
2	+ Direct noise	56.47	26.51	46.04	54.46	38.85	5.33	17.21
3	+ Back-translation	**58.17**	27.87	**47.77**	54.84	**39.78**	6.22	19.14
4	Transformer	39.92	39.13	39.75	55.39	30.97	9.23	21.05
5	+ Direct noise	41.89	38.02	41.05	55.60	31.52	9.07	21.08
6	+ Back-translation	43.42	**40.92**	42.88	**56.13**	33.34	**10.53**	**23.24**

パスとドメインが異なるためであると考えられ，英語文法誤り訂正でドメインが異なるとモデルの性能が低下すると報告した Flachs ら [42] の実験と整合性のある結果である．

次に各文法誤り訂正モデルの誤りタイプ別の性能を表 5 に示す．誤りタイプは TEC-JL 中の割合が 5% 以上の誤りタイプのみを記載した．また誤りタイプ別の性能には Recall を用いた[*14]．表 5 より，CNN と Transformer の両方とも綴り誤りの性能が高いことが分かる．また助詞誤りの性能も高く，Transformer では 50% 近くの助詞誤りを訂正できている．したがって綴り誤りや助詞誤りは日本語文法誤り訂正モデルにとって比較的訂正しやすい誤りタイプであると考えられる．綴り誤りや助詞誤りが訂正しやすい理由には様々な要素があるが，例えば学習者が犯しやすい誤りのため訓練データに同様の誤りが多く存在することなどが挙げられる．一方名詞誤りは各モデルの訂正性能が比較的低く，CNN では約 8% であり Transformer でも約 18% である．名詞誤りは文脈を考慮して訂正する必要が多いため，誤りタイプの中でも訂正性能が低くなった．英語文法誤り訂正でも名詞誤りの訂正性能が低いことは指摘されており [14]，名詞誤りの訂正性能向上は日本語と英語で共通の課題である．

6. 擬似データの利用

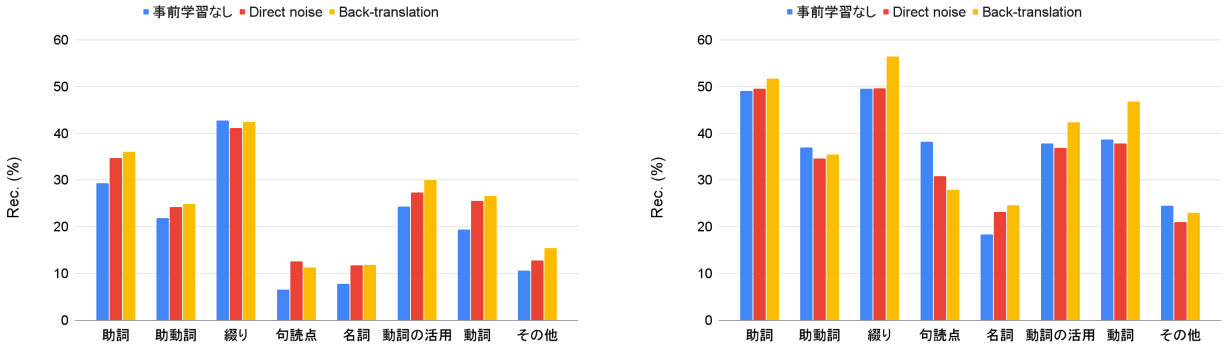
現在文法誤り訂正で主流のエンコーダ・デコーダモデル [43], [44] は大量の訓練データを必要とする [45]．しかし文法誤り訂正に利用可能な学習者データは限られている．そのため英語文法誤り訂正では擬似データを活用し，エンコーダ・デコーダモデルの性能向上を図ることが標準的になっている [3], [39], [41], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56]．ただし日本語文法誤り訂正では，擬似データがエンコーダ・デコーダモデルの性能に与える影響があまり明らかになっていない．そこで本節では，既存の擬似データ生成手法を適用し，日本語文法誤り訂正に擬似データを用いた場合のモデルの性能変化を観察する．

6.1 実験設定

エンコーダ・デコーダモデルには 5 節と同様，CNN と Transformer を使用する．各モデルのアーキテクチャや評価方法は 5 節と同様である．擬似データ生成時の生成元コーパスには Wiki-40B [57] の日本語部分から抽出した 1,000 万文を用いた．Kiyono ら [41] に従い，擬似データは事前学習で使用し，学習者データはファインチューニングにのみ用いた．また事前学習及びファインチューニング時の設定も Kiyono ら [41] に従った．使用する擬似データ生成手法は以下の通りである．

Direct noise [3] 原文に 4 つの編集操作（削除，挿入，置換，シャッフル）を行う．具体的には原文をトークンに区切り，次の操作を確率的に適用する．(i) 削除: 10%の確率でトークンを削除する．(ii) 挿入: 10%の確率でトークンの後ろに生成元コーパス中のトークンを挿入する．(iii) 置換: 10%の確率でトークンを生成元コーパス中のトークンに置き換える．(iv) シャッフル: 各トークンの位置番号に対し正規分布から抽出した値を加え，昇順に整列する．本実験では Zhao ら [3]

[*14] 誤りタイプ別の性能を $F_{0.5}$ で評価するにはモデル出力に誤用タグを付与し，誤りタイプ別の Precision を求める必要がある．英語では ERRANT で誤用タグを自動推定し，誤りタイプ別の Precision を求めることが可能である．しかし日本語では公開された誤用タグ付けツールがないため Recall で評価した．



(a) CNN **(b)** Transformer

図 2: 擬似データを利用した場合の各文法誤り訂正モデルの Recall (%).

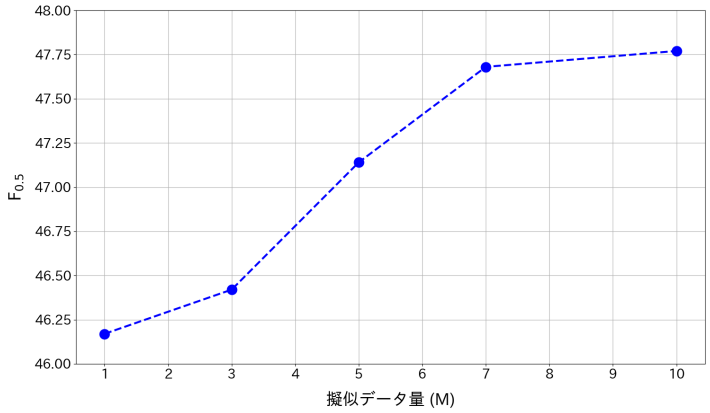


図 3: TEC-JL における擬似データ量を変化させた時の $F_{0.5}$.

が公開しているスクリプトを使用した[*15].

Back-translation [46] 逆翻訳モデル [58] を用いて原文に誤りを生成させる．逆翻訳モデルとは入力と出力を入れ替えて訓練したモデルである．文法誤り訂正の場合，訂正文を入力とし学習者文を出力として訓練したモデルになる．Xie ら [46] はビームサーチ時，毎ステップの各仮説のスコアに $r\beta_{\text{random}}$ をノイズとして加え，より多様な誤りを含むように改良した．ここで，r は区間 $[0, 1]$ の一様分布からランダムに選択される値であり，β_{random} はノイズの大きさを調節するためのハイパーパラメータである．本実験では Transformer を逆翻訳モデルに用い，事前実験において開発データ上で最大の GLEU$^+$ スコアとなった時の値である $\beta_{\text{random}} = 8$ を使用した．

6.2 実験結果

表 6 に擬似データを使用した場合の各文法誤り訂正モデルの性能を示す．表 6 より，擬似データを使用することで各モデルの性能が向上したことが分かる．TEC-JL では，CNN に Back-translation で生成した擬似データを用いた場合に $F_{0.5}$ が最も高くなった．FLUTEC と NAIST 誤用コーパスでは，Transformer に Back-translation を使用した場合にそれぞれ GLEU$^+$ スコアと $F_{0.5}$ が最も高くなった．全体的な傾向としては，Direct noise よりも Back-translation を用いた方がモデルの性能が向上している．この理由には Back-translation の方がより多様な誤りを生成できることが考えられる．

図 2 に擬似データで事前学習した場合の各文法誤り訂正モデルの誤りタイプ別の性能を示す．ここで事前学習なしは擬似データを用いず学習者データのみで訓練したモデルの性能を表す．図 2 より，両方の擬似データ生成手法で助詞誤りの訂正性能が事前学習なしの場合のよりも向上していることが分かる．助詞誤りは学習者が犯しやすい誤りタイプであるため [16]，両方の擬似データ生成手法で性能が向上したことは好ましい結果である．また Transformer では，Back-translation を使用した場合に綴り誤りの訂正性能が向上した．しかし CNN では，綴り誤りの訂正性能が Back-translation を使用しても向上しなかった．動詞の活用誤りの訂正性能は，CNN では両方の擬似データ生成手法で性能が向上しているが，Transformer では Direct noise で性能が低下した．したがってどのような擬似データ生成手法が有効かはモデルのアーキテクチャに依存することが示唆され，今後より詳細な分析が必要である．

[*15] https://github.com/yuantiku/fairseq-gec

表 7: TEC-JL における擬似データ量（文数）を変化させた時の誤りタイプ別の Recall (%)．Δ は擬似データなしの性能からの各擬似データ量での差分を表す．

	誤りタイプ							
擬似データ量	助詞	Δ	助動詞	Δ	綴り	Δ	句読点	Δ
なし	29.34	-	21.91	-	42.79	-	6.57	-
1M	33.46	4.12	22.67	0.76	44.46	1.67	11.08	4.51
3M	36.14	6.80	23.37	1.46	43.42	0.63	10.68	4.11
5M	37.07	7.73	24.54	2.63	44.19	1.40	11.42	4.85
7M	37.37	8.03	26.39	4.48	45.02	2.23	9.51	2.94
10M	36.03	6.69	24.85	2.94	42.49	−0.30	11.29	4.72
	名詞	Δ	動詞の活用	Δ	動詞	Δ	その他	Δ
なし	7.80	-	24.30	-	19.45	-	10.62	-
1M	10.57	2.77	27.75	3.45	24.05	4.60	13.52	2.90
3M	11.67	3.87	30.15	5.85	25.30	5.85	13.27	2.65
5M	10.50	2.70	28.34	4.04	28.37	8.92	14.19	3.57
7M	11.31	3.51	33.09	8.79	28.28	8.83	15.06	4.44
10M	11.89	4.09	30.04	5.74	26.57	7.12	15.49	4.87

6.3 擬似データ量が性能に与える影響

擬似データ量が訂正性能に与える影響をより詳細に調べるため，擬似データ量を変化させた時のモデルの性能を調査した．具体的には擬似データ量を 1M，3M，5M，7M，10M と変化させた時の性能を調査した．図 3 に TEC-JL における擬似データ量を変化させた時の $F_{0.5}$ を示す．本実験では，表 6 で TEC-JL において最高性能を達成した CNN+Back-translation モデルを用いた．図 3 より，擬似データ量の増加とともにモデルの性能は向上することが分かる．これは英語文法誤り訂正での Kiyono ら [41] や Wan ら [52] の報告と整合性のある結果である．次に誤りタイプ別の性能はどのように変化するかを調べる．表 7 に擬似データ量を変化させた時の誤りタイプ別の Recall を示す．表 7 より，ほとんどの場合，擬似データなしの時に比べ擬似データを使用した時の方が誤りタイプ別の性能も向上することが分かる．また助詞誤りや助動詞誤りなどは擬似データ量が 1M の時と比べ 10M では Recall が 2% 程度上昇している．一方綴り誤りでは擬似データなしの時よりも 10M の時の方が性能が下がった．また句読点誤りでは擬似データ量が 1M と 10M の時で訂正性能がほとんど変わっていない．Back-translation では綴り誤りや句読点誤りを上手く生成できていないことが考えられ，擬似データの生成方法によっても性能が向上しやすい誤りタイプとそうでない誤りタイプが存在することが分かった．

7. おわりに

本研究では日本語文法誤り訂正のための評価コーパス TEC-JL へ誤用タグを付与した．そして誤用タグを利用し，日本語文法誤り訂正モデルの誤りタイプ別の性能を調査した．今後の課題としては誤りタイプ別の性能を $F_{0.5}$ で評価することが挙げられる．本実験では誤りタイプ別の評価に Recall を使用しており，$F_{0.5}$ では評価できていない．誤りタイプ別の $F_{0.5}$ を求めるには，モデル出力に誤用タグを付与し誤りタイプ別の Precision を求める必要がある．しかし日本語では ERRANT のような誤用タグ付けツールが存在せず，人手で誤用タグを付与するのはコストが高い．したがって日本語でも誤用タグ付けツールを開発し，モデルの誤りタイプ別の性能を $F_{0.5}$ で評価可能にすることが今後の課題である．

謝辞 Lang-8 のデータを提供してくださった株式会社 Lang-8 の喜洋洋氏に感謝申しあげます．

参考文献

[1] Yuan, Z. and Briscoe, T.: Grammatical error correction using neural machine translation, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 380–386 (2016).

[2] Chollampatt, S. and Ng, H. T.: A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pp. 5755–5762 (2018).

[3] Zhao, W., Wang, L., Shen, K., Jia, R. and Liu, J.: Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Lin-*

guistics: Human Language Technologies, pp. 156–165 (2019).

[4] Kaneko, M., Mita, M., Kiyono, S., Suzuki, J. and Inui, K.: Encoder-Decoder Models Can Benefit from Pre-trained Masked Language Models in Grammatical Error Correction, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4248–4254 (2020).

[5] Rothe, S., Mallinson, J., Malmi, E., Krause, S. and Severyn, A.: A Simple Recipe for Multilingual Grammatical Error Correction, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 702–707 (2021).

[6] Yuan, Z., Taslimipoor, S., Davis, C. and Bryant, C.: Multi-Class Grammatical Error Detection for Correction: A Tale of Two Systems, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8722–8736 (2021).

[7] Sun, X. and Wang, H.: Adjusting the Precision-Recall Trade-Off with Align-and-Predict Decoding for Grammatical Error Correction, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 686–693 (2022).

[8] Schneider, D. and McCoy, K. F.: Recognizing Syntactic Errors in the Writing of Second Language Learners, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pp. 1198–1204 (1998).

[9] Dahlmeier, D. and Ng, H. T.: Grammatical Error Correction with Alternating Structure Optimization, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 915–923 (2011).

[10] Gehring, J., Auli, M., Grangier, D., Yarats, D. and Dauphin, Y. N.: Convolutional Sequence to Sequence Learning, *Proceedings of the 34th International Conference on Machine Learning*, pp. 1243–1252 (2017).

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is All you Need, *Advances in Neural Information Processing Systems 30*, pp. 5998–6008 (2017).

[12] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H. and Bryant, C.: The CoNLL-2014 Shared Task on Grammatical Error Correction, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14 (2014).

[13] 三田雅人，水本智也，金子正弘，永田　亮，乾健太郎：文法誤り訂正モデルの横断評価，自然言語処理，Vol. 28, No. 1, pp. 160–182 (2021).

[14] Bryant, C., Felice, M., Andersen, Ø. E. and Briscoe, T.: The BEA-2019 Shared Task on Grammatical Error Correction, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75 (2019).

[15] Koyama, A., Kiyuna, T., Kobayashi, K., Arai, M. and Komachi, M.: Construction of an Evaluation Corpus for Grammatical Error Correction for Learners of Japanese as a Second Language, *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 204–211 (2020).

[16] 大山浩美，小町　守，松本裕治：日本語学習者の作文における誤用タイプの階層的アノテーションに基づく機械学習による自動分類，自然言語処理，Vol. 23, No. 2, pp. 195–225 (2016).

[17] 井上　優，宇佐美洋，成田高宏，鑓水兼高：作文対訳データベースの多様な利用のために:「日本語教育のための言語資源及び学習内容に関する調査研究」成果報告書，国立国語研究所. (2006).

[18] Felice, M., Bryant, C. and Briscoe, T.: Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments, *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 825–835 (2016).

[19] Bryant, C., Felice, M. and Briscoe, T.: Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 793–805 (2017).

[20] de Marneffe, M.-C., Manning, C. D., Nivre, J. and Zeman, D.: Universal Dependencies, *Computational Linguistics*, Vol. 47, No. 2, pp. 255–308 (2021).

[21] 浅原正幸，金山　博，宮尾祐介，田中貴秋，大村　舞，村脇有吾，松本裕治：Universal Dependencies 日本語コーパス，自然言語処理，Vol. 26, No. 1, pp. 3–36 (2019).

[22] Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R. and Zeman, D.: Universal Dependencies v1: A Multilingual Treebank Collection, *Proceedings of the 10th Language Resources and Evaluation Conference*, pp. 1659–1666 (2016).

[23] Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., Schuster, S., Tyers, F. and Zeman, D.: Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection, *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4034–4043 (2020).

[24] Honnibal, M. and Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017).

[25] Boyd, A.: Using Wikipedia Edits in Low Resource Grammatical Error Correction, *Proceedings of the 4th Workshop on Noisy User-generated Text*, pp. 79–84 (2018).

[26] Cotet, T.-M., Ruseti, S. and Dascalu, M.: Neural Grammatical Error Correction for Romanian, *Proceedings of the 32nd IEEE International Conference on Tools with Artificial Intelligence*, pp. 625–631 (2020).

[27] Korre, K., Chatzipanagiotou, M. and Pavlopoulos, J.: ELERRANT: Automatic Grammatical Error Type Classification for Greek, *Proceedings of the 2021 International Conference on Recent Advances in Natural Language Processing*, pp. 708–717 (2021).

[28] Náplava, J., Straka, M., Straková, J. and Rosen, A.: Czech Grammar Error Correction with a Large and Diverse Corpus, *Transactions of the Association for Computational Linguistics*, Vol. 10, pp. 452–467 (2022).

[29] 水本智也，小町　守，永田昌明，松本裕治：日本語学習者の作文自動誤り訂正のための語学学習 SNS の添削ログからの知識獲得，人工知能学会論文誌，Vol. 28, No. 5, pp. 420–432 (2013).

[30] 木山　朔，上坂奏人，佐藤郁子，佐藤京也，米田悠人，小山碧海，三田雅人，岡　照晃，小町　守：日本語文法誤り訂正の流暢性評価に向けたデータ作成，言語処理学会第 28 回年次大会発表論文集，pp. 1704–1709 (2022).

[31] 水本智也：日本語文法誤り訂正における最適な分割単位の調査，言語処理学会第 26 回年次大会発表論文集，pp. 1336–1339 (2020).

[32] Kudo, T.: Subword Regularization: Improving Neural Network Translation Models with Multiple Subword

Candidates, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 66–75 (2018).

[33] Kudo, T. and Richardson, J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71 (2018).

[34] Sakaguchi, K., Napoles, C., Post, M. and Tetreault, J.: Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality, *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 169–182 (2016).

[35] Dahlmeier, D. and Ng, H. T.: Better Evaluation for Grammatical Error Correction, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 568–572 (2012).

[36] Napoles, C., Sakaguchi, K., Post, M. and Tetreault, J.: GLEU Without Tuning, *arXiv preprint arXiv:1605.02592 [cs.CL]* (2016).

[37] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318 (2002).

[38] Ge, T., Wei, F. and Zhou, M.: Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study, *arXiv preprint arXiv:1807.01270 [cs.CL]* (2018).

[39] Lichtarge, J., Alberti, C., Kumar, S., Shazeer, N., Parmar, N. and Tong, S.: Corpora Generation for Grammatical Error Correction, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3291–3301 (2019).

[40] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D. and Auli, M.: fairseq: A Fast, Extensible Toolkit for Sequence Modeling, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 48–53 (2019).

[41] Kiyono, S., Suzuki, J., Mizumoto, T. and Inui, K.: Massive Exploration of Pseudo Data for Grammatical Error Correction, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 2134–2145 (2020).

[42] Flachs, S., Lacroix, O., Yannakoudakis, H., Rei, M. and Søgaard, A.: Grammatical Error Correction in Low Error Density Domains: A New Benchmark and Analyses, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 8467–8478 (2020).

[43] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems 27*, pp. 3104–3112 (2014).

[44] Bahdanau, D., Cho, K. and Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate, *Proceedings of the 3rd International Conference on Learning Representations* (2015).

[45] Koehn, P. and Knowles, R.: Six Challenges for Neural Machine Translation, *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39 (2017).

[46] Xie, Z., Genthial, G., Xie, S., Ng, A. and Jurafsky, D.: Noising and Denoising Natural Language: Diverse Back-translation for Grammar Correction, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 619–628 (2018).

[47] Ge, T., Wei, F. and Zhou, M.: Fluency Boost Learning and Inference for Neural Grammatical Error Correction, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1055–1065 (2018).

[48] Grundkiewicz, R., Junczys-Dowmunt, M. and Heafield, K.: Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 252–263 (2019).

[49] Lichtarge, J., Alberti, C. and Kumar, S.: Data Weighted Training Strategies for Grammatical Error Correction, *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 634–646 (2020).

[50] Wang, L. and Zheng, X.: Improving Grammatical Error Correction Models with Purpose-Built Adversarial Examples, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 2858–2869 (2020).

[51] Zhou, W., Ge, T., Mu, C., Xu, K., Wei, F. and Zhou, M.: Improving Grammatical Error Correction with Machine Translation Pairs, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 318–328 (2020).

[52] Wan, Z., Wan, X. and Wang, W.: Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation, *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2202–2212 (2020).

[53] Stahlberg, F. and Kumar, S.: Synthetic Data Generation for Grammatical Error Correction with Tagged Corruption Models, *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 37–47 (2021).

[54] Yasunaga, M., Leskovec, J. and Liang, P.: LM-Critic: Language Models for Unsupervised Grammatical Error Correction, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7752–7763 (2021).

[55] Li, X. and He, J.: Data Augmentation of Incorporating Real Error Patterns and Linguistic Knowledge for Grammatical Error Correction, *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 223–233 (2021).

[56] 古山翔太，高村大也，岡崎直観：多様な規則を活用した文法誤り訂正のデータ拡張に関する分析，自然言語処理，Vol. 29, No. 2, pp. 542–586 (2022).

[57] Guo, M., Dai, Z., Vrandečić, D. and Al-Rfou, R.: Wiki-40B: Multilingual Language Model Dataset, *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 2440–2452 (2020).

[58] Sennrich, R., Haddow, B. and Birch, A.: Improving Neural Machine Translation Models with Monolingual Data, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 86–96 (2016).