日本語文法誤り訂正の流暢性評価に向けたデータ作成

木山朔 [1]　上坂奏人 [1]　佐藤郁子 [1]　佐藤京也 [1]　米田悠人 [1]
小山碧海 [1]　三田雅人 [2,1]　岡照晃 [1]　小町守 [1]
[1] 東京都立大学　[2] 理化学研究所
{kiyama-hajime, uesaka-minato, sato-ayako, sato-keiya, yoneda-yuto,
koyama-aomi}@ed.tmu.ac.jp, masato.mita@riken.jp,
{teruaki-oka, komachi}@tmu.ac.jp

概要

本研究では，日本語文法誤り訂正の流暢性評価データを作成する．既存の日本語文法誤り訂正の評価データでは，文法的に正しくなる最小限の編集に基づき訂正文が作成されている．しかし最小限の編集に基づく訂正文は文法的ではあるものの流暢性に欠ける．訂正文の流暢性が欠けた評価データに依存してモデルを開発すると，文法的かつ流暢な出力をするモデルが正当に評価されない．そこで文法的なだけでなく流暢な編集を行った評価データを作成し，モデルの流暢性も含めた評価を可能にする．

1　はじめに

文法誤り訂正とは，語学学習者が書いた文章中の誤りを計算機が自動で訂正するタスクである[1)]．これまでの文法誤り訂正は学習者コーパスが豊富な英語を中心に研究されてきた [1, 2, 3, 4, 5, 6, 7, 8]．しかし，ドイツ語やロシア語などでも学習者コーパスが整備され [9, 10, 11]，英語以外でも研究が盛んになっており [12, 13, 14, 15, 16]．日本語を対象とした研究も広がりを見せている [17, 18, 19, 20, 21, 22]．

日本語文法誤り訂正では，Koyama ら [21] が文法的に正しくなる最小限の編集（**Minimal edits**）に基づく評価データを作成した．表 1 の 2 行目に，彼らの Minimal edits に基づく訂正例を示す．Minimal edits では学習者文中の “時頃” を “頃” に書き換えることで，文法的に正しくなる最小限の訂正を行っている．Minimal edits は NUCLE [23] という英語学習者コーパスにも使用されており，NUCLE の一部である CoNLL-2014 shared task [24] の評価データは英語文法誤り訂正で最も代表的な評価データである．

表 1: Minimal edits 及び Fluency edits に基づく訂正例．

学習者文	私が**ハイスクール**を**終える**べき**時頃**です。
Minimal	私がハイスクールを終えるべき**頃**です。
Fluency	私が**高校**を**卒業する**べき**頃**です。

一方，英語文法誤り訂正では Minimal edits に基づく訂正文が母語話者にとって文法的ではあるものの流暢性の低い文になっていると指摘されている [25]．また Sakaguchi ら [25] は文法誤り訂正の目的を文法的なだけでなく母語話者の流暢さを持つ文章の作成へと根本的にシフトすべきと提唱している．そして彼らの提唱を受け，文法的なだけでなく流暢な編集（**Fluency edits**）を行った評価データが公開され [26]，新たなベンチマークデータとしてコミュニティに享受される形で英語文法誤り訂正の研究が進められている．

Minimal edits に基づく訂正文の流暢性が低いという問題は日本語でも同様である．例えば表 1 の場合，日本語母語話者にとっては，“時頃” を “頃” に訂正するだけでなく，“ハイスクール” を “高校” に，“終える” を “卒業する” に訂正する方がより流暢であると考えられる．しかし日本語文法誤り訂正では Fluency edits ベースの評価データは未だ作成されていない．またそのため，流暢な出力をするモデルが正当に評価されていない．

そこで，我々は日本語文法誤り訂正のための流暢性評価データを作成する[2)]．具体的には，Lang-8 コーパス [19] 中の日本語学習者文に対し，文法的なだけでなく流暢な編集を行った訂正文を人手で付与する．またベースラインとして 3 つの日本語文法誤り訂正モデルを用意し，それらのモデルを流暢性の観点から評価した結果を報告する．

1)　本稿では，便宜上，文法誤りだけでなく綴りや語彙選択の誤りの訂正も文法誤り訂正に含める．

2)　作成した評価データは以下のサイトで公開予定である．
https://github.com/kiyama-hajime/FLUTEC

表 2: 各訂正基準に基づく訂正例.

訂正基準	学習者文	訂正文
R1	このために、日本人の学生はもう学生生活を**準備してけど**、オーストラリア人の学生はもう作業生活の準備します。	このために、日本人の学生は学生生活の**準備をしていますが**、オーストラリア人の学生はもう社会人生活の準備をします。
R2	私は日本に行ったら、ドラッグがあんまり使わないことを見なかった、だってここで大きい「問題」ようだ。	私が日本に行ったら、ドラッグがあんまり使われることを見なかった。だってここでは大きい「問題」のようだ。

2 関連研究

日本語教育で活用されている日本語学習者コーパスには作文対訳データベース[3]や日本語学習者作文コーパス[4]などがある．作文対訳データベースでは，学習者の手書き作文に対する日本語教師の訂正情報が収録されている．日本語学習者作文コーパスでは，初級から上級までの中国語や韓国語を母語とする学習者の作文データ及び訂正情報が収録されている．国際日本語学習者作文コーパス及び誤用辞典[5]やなたね[6]でも同様に，学習者作文に対する訂正情報が収録されているが，これらの元データは一般には公開されていないため日本語文法誤り訂正モデルの評価に使用できない．日本語文法誤り訂正モデルの評価を目的に公開されている学習者コーパスとして TEC-JL [21] がある．TEC-JL では，Lang-8 コーパス中の日本語学習者文に Minimal edits ベースの訂正文を付与している．我々は，TEC-JL と同様に Lang-8 コーパス中の学習者文を人手で訂正し公開するが，訂正方針は Minimal edits ではなく Fluency edits に基づいた訂正を採用する．

英語文法誤り訂正で代表的な学習者コーパスに NUCLE [23] がある．NUCLE はシンガポールの大学生が書いた作文を英語教師が訂正した英語学習者コーパスである．Minimal edits に基づき訂正されており，NUCLE の一部は CoNLL-2014 shared task の評価データに使用されている．一方，JFLEG [26, 27] は Fluency edits に基づき作成された英語学習者コーパスである．JFLEG では，母語や習熟度の異なる学習者が書いた文に対し，クラウドソーシングを用いて訂正文が付与されている．我々は，JFLEG と同様に Fluency edits に基づくコーパスを作成するが，英語ではなく日本語学習者コーパスを作成する．

3） https://mmsrv.ninjal.ac.jp/essay
4） http://sakubun.jpn.org
5） https://corpus.icjs.jp/corpus_ja
6） https://hinoki-project.org/natane

3 流暢性評価データの作成

3.1 利用した言語資源と前処理

我々は TEC-JL と同様に，Lang-8 コーパス中の学習者文に再アノテーションを行う．Lang-8 コーパスでは学習者が書いた作文が文章単位で収録されている．また各文章は自動で文分割されており，文ごとに訂正文が付与されている．さらに各文章に対し，母語と学習言語，各文章を識別するためのユニークな番号（ジャーナル ID）が付与されている．

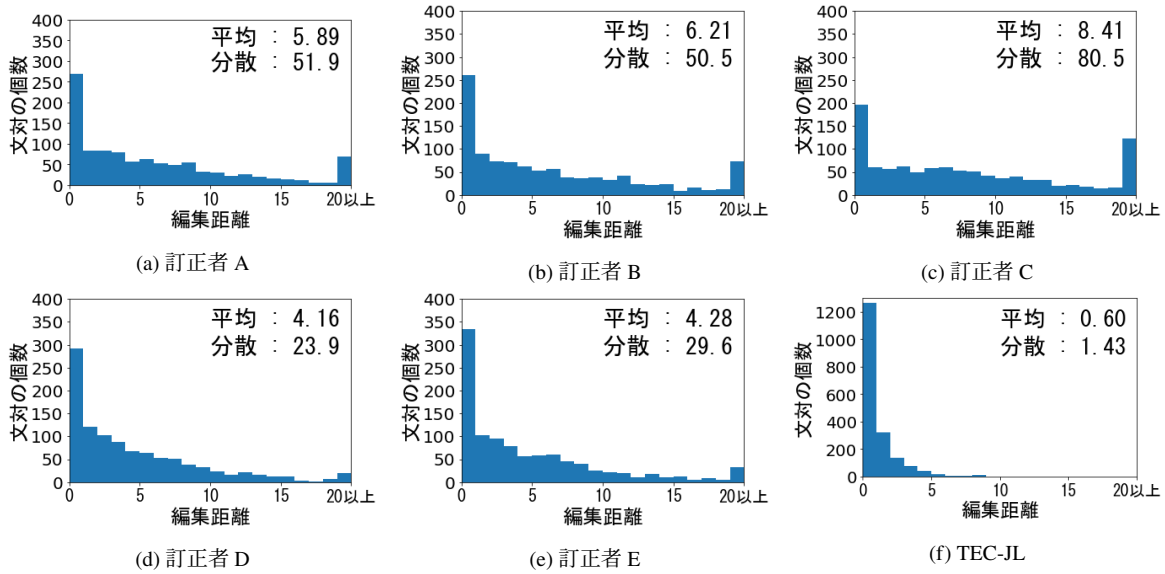
本研究で作成する検証データ及び評価データに用いる日本語学習者文は以下の手順で用意した．

Step 1. Lang-8 コーパスから，日本語を学習言語とする文章を抽出する[7]．
Step 2. 検証データと評価データがそれぞれ 1,000 文程度になるように文章を選択する[8]．
Step 3. 検証データや評価データとして不適切な文を人手で取り除き[9]，文分割に失敗している箇所を人手で正しい分割に修正する．

3.2 訂正基準

流暢性を考慮した評価データを作成するために，3.1 節で用意した学習者文に対し，日本語母語話者 5 人が文法的に正しく，かつ流暢な訂正を行う．また訂正の際は，原文の意味を保持するように訂正する．さらに TEC-JL と同様に，訂正は文章単位で行う．流暢な訂正には常識に関する誤りの訂正 [28] も含み，元の文章に対して複数の解釈が存在する場合は訂正者各自が文章全体を見て最尤な訂正を行う．

7） ただし TEC-JL 中の文章との重複を避けるため，TEC-JL に使用された文章と同じジャーナル ID の文章は抽出しない．
8） この時，検証データ及び評価データ中の学習者の母語の分布が，Lang-8 コーパス中の日本語学習者の母語の分布と同様になるように文章を選択した．
9） 例えば，日本語以外の言語で書かれている文や記号のみの文を取り除いた．



(a) 訂正者 A　(b) 訂正者 B　(c) 訂正者 C

(d) 訂正者 D　(e) 訂正者 E　(f) TEC-JL

図 1: 本研究及び TEC-JL における学習者文と訂正文間の編集距離の分布.

また評価データ全体で一貫した訂正を行うため，以下の訂正基準を設定した．表 2 に各訂正基準に基づく訂正例を示す．

R1. 1 文章内では漢字やカタカナなどの表記，常体や敬体を統一する．

R2. 一文を複数文に分割する訂正はしてもよいが，複数文を一文に統合する訂正はしない．

R1 は表記や文体を統一することで流暢性が向上すると考え設定した．R2 は，複数の参照文がある場合，複数文を一文に統合するような訂正の自動評価が困難であるため設定した．

3.3 分析

定量評価 図 1 に各訂正者の学習者文と訂正文間の文字単位の編集距離の分布を，TEC-JL での編集距離の分布とともに示す．訂正者 C は編集距離の平均が最も高く，大幅な訂正を最も多く行っていることが分かる．一方で，訂正者 D と訂正者 E は編集距離の平均が低く，他の訂正者と比べて大幅な訂正をあまり行っていない．TEC-JL と比較すると，本研究で作成した評価データの方が編集距離の平均が高く，流暢な訂正文にするために大幅に訂正していることが分かる．また付録 A に訂正者間の文単位の類似度に関する分析を記載する．

定性評価 表 3 に，各訂正者の流暢性を考慮した訂正文の例を示す．それぞれの訂正者は，“速く０００番に携帯で手伝える” の部分を主に訂正しており，訂正方法には様々なバリエーションがあることが分かる．また訂正者 B は，より流暢な文にするため，“緊急の場合は” の部分も含めて訂正している．このように訂正者ごとに使用する表現が分かれたり，文法的に正しい部分も含めて訂正したりすることがある．さらに常識に関する誤りとして，“０００番” をそのままにするか，“１１０番” に訂正するかの判断が分かれている．この違いはそれぞれの訂正者が学習者の居住地をどのように推測したかの違いである．“０００番” のままにしている訂正者はオーストラリアなどの緊急時の番号が 000 である国に住んでいると推測し，“１１０番” にした訂正者は日本に住んでいると推測したことにより，このような違いが発生した．

4 実験

本節では日本語文法誤り訂正モデルの流暢性を評価するため，機械翻訳ベースの 3 つの文法誤り訂正モデルの性能を比較する．

4.1 実験設定

データセット 訓練データには Lang-8 コーパスを用いる．ただし，TEC-JL と本研究で作成した評価データに含まれるジャーナル ID を持つ文対は取り除き，さらに文長制限などのノイズ除去を行った．水本 [29] に従い，学習者文は文字に分割し，訂正文は語彙サイズ 16,000 で SentencePiece[10] [30] によるトークン化を行った．検証データには，本研究で作成した検証データを用いる．本実験で使用した

10) https://github.com/google/sentencepiece

表 3: 各訂正者の流暢性を考慮した訂正文の例.

学習者文	ちなみに、緊急の場合は、速く０００番に携帯で手伝える。
訂正者 A	ちなみに、緊急の場合は、携帯に１１０番を押して連絡できる。
訂正者 B	ちなみに、緊急電話は、０００番で素早くつながる。
訂正者 C	ちなみに、緊急の場合は、携帯の０００番で助けを求められる。
訂正者 D	ちなみに、緊急の場合は、素早く１１０番に携帯で伝える。
訂正者 E	ちなみに、緊急の場合は、携帯で０００番にかけるとよい。

データセットの詳細は付録 B に記載する.

性能評価 評価データには TEC-JL 及び本研究で作成した評価データ[11]を用いる. Minimal edits ベースの評価データである TEC-JL は既存研究で広く使われている M^2 scorer [31] で評価する. 一方 Fluency edits の場合 Minimal edits と比べ原文と参照文間のアライメントが取りづらいことから, 本研究で作成した評価データはアライメントをとる必要がない GLEU [32] で評価する. 両方の評価データにおいて, 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, 文字単位で評価する.

文法誤り訂正モデル 文法誤り訂正モデルには Koyama ら [21] が使用している SMT [33] と CNN [34] に加え, Transformer [35] を用いる. 各モデルの設定は以下の通りである.

SMT Moses [12] [33] を実装に使用した. GIZA++ [13] [36] を対応づけツールに使用し, KenLM [37] を用いて 3-gram 言語モデルを構築した. 言語モデルの訓練には訓練データ中の訂正文を用いた. また, 検証データを用いて BLEU [38] を最大化するように MERT [39] を行った.

CNN fairseq [14] [40] を実装に使用した. アーキテクチャは Chollampatt ら [41] と同様である. また, 訓練時の最適化方法や推論時の設定は Kiyono ら [42] に従った.

Transformer fairseq を実装に使用した. アーキテクチャは Vaswani ら [35] の “Transformer (base)” と同様である. また, 訓練時の最適化方法や推論時の設定は Kiyono ら [42] に従った.

4.2 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出力とみなして評価した時のスコアである. また人間と文法誤り訂正モデルの性能差を調べるため, 交差検証のように, 各参照文を出力とみなしそれ以外の参照文で評価した時のスコアの平均を人間のスコアとして求めた[15]. TEC-JL では, CNN や Transformer よりも SMT の方が Precision が高い. 一方, Recall は Transformer の方が高いことから, Transformer はより多くの訂正をしていると考えられる. また本研究で作成した評価データでは, 3 つのモデルの内, Transformer のスコアが最も高いことから, Transformer は他のモデルよりも比較的流暢な訂正をしていると考えられる. 人間と各モデルのスコアを比較すると, 全てのモデルのスコアは人間よりも低く, 改善の余地が残されていることが分かる. 各モデルの出力例を付録 D に記載する.

表 4: 各文法誤り訂正モデルの性能.

	TEC-JL			Ours
文法誤り訂正モデル	Prec.	Rec.	$F_{0.5}$	GLEU
訂正なし	-	-	-	52.5
人間	63.0	58.9	62.1	62.6
SMT	**47.1**	7.9	23.7	53.5
CNN	41.8	16.9	**32.3**	55.0
Transformer	30.0	**27.1**	29.3	**56.2**

5 おわりに

本研究では日本語文法誤り訂正のための流暢性評価データを作成した. 本評価データでは, 学習者文に対し Fluency edits に基づく訂正を行っている. さらに, 1 つの学習者文につき 5 つの訂正文を付与しており, 多様な訂正をカバーしている. 今後の課題としては, Lang-8 コーパス以外のドメインに対しても Fluency edits ベースの日本語学習者コーパスを作成することが挙げられる.

謝辞

Lang-8 のデータ使用に際して, 株式会社 Lang-8 の喜洋洋氏に感謝申しあげます.

11) 本稿執筆時点では評価データ及び検証データ全体が完成しておらず, 作成途中のものを使用した.
12) https://github.com/moses-smt/mosesdecoder
13) https://github.com/moses-smt/giza-pp
14) https://github.com/pytorch/fairseq
15) 人間のスコアと比較可能にするため, 各モデル及び訂正なしの性能も人間のスコアと同様に測定した. 全ての参照文で評価した時の各モデルの性能は付録 C に記載する.

参考文献

[1] Chris Brockett, William B. Dolan, and Michael Gamon. Correcting ESL Errors Using Phrasal SMT Techniques. In **COLING-ACL**, pp. 249–256, 2006.

[2] Rachele De Felice and Stephen G. Pulman. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In **COLING**, pp. 169–176, 2008.

[3] Daniel Dahlmeier and Hwee Tou Ng. Grammatical Error Correction with Alternating Structure Optimization. In **ACL**, pp. 915–923, 2011.

[4] Marcin Junczys-Dowmunt and Roman Grundkiewicz. Phrase-based Machine Translation is State-of-the-Art for Automatic Grammatical Error Correction. In **EMNLP**, pp. 1546–1556, 2016.

[5] Roman Grundkiewicz and Marcin Junczys-Dowmunt. Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation. In **NAACL**, pp. 284–290, 2018.

[6] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. Corpora Generation for Grammatical Error Correction. In **NAACL**, pp. 3291–3301, 2019.

[7] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. Encoder-Decoder Models Can Benefit from Pre-trained Masked Language Models in Grammatical Error Correction. In **ACL**, pp. 4248–4254, 2020.

[8] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LM-Critic: Language Models for Unsupervised Grammatical Error Correction. In **EMNLP**, pp. 7752–7763, 2021.

[9] Adriane Boyd. Using Wikipedia Edits in Low Resource Grammatical Error Correction. In **W-NUT**, pp. 79–84, 2018.

[10] Alla Rozovskaya and Dan Roth. Grammar Error Correction in Morphologically Rich Languages: The Case of Russian. **TACL**, Vol. 7, pp. 1–17, 2019.

[11] Viet Anh Trinh and Alla Rozovskaya. New Dataset and Strong Baselines for the Grammatical Error Correction of Russian. In **Findings: ACL-IJCNLP**, pp. 4103–4111, 2021.

[12] Jakub Náplava and Milan Straka. Grammatical Error Correction in Low-Resource Scenarios. In **W-NUT**, pp. 346–356, 2019.

[13] Satoru Katsumata and Mamoru Komachi. Stronger Baselines for Grammatical Error Correction Using a Pretrained Encoder-Decoder Model. In **AACL-IJCNLP**, pp. 827–832, 2020.

[14] Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova, and Mamoru Komachi. Cross-lingual Transfer Learning for Grammatical Error Correction. In **COLING**, pp. 4704–4715, 2020.

[15] Simon Flachs, Felix Stahlberg, and Shankar Kumar. Data Strategies for Low-Resource Grammatical Error Correction. In **BEA**, pp. 117–122, 2021.

[16] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A Simple Recipe for Multilingual Grammatical Error Correction. In **ACL-IJCNLP**, pp. 702–707, 2021.

[17] Hisami Suzuki and Kristina Toutanova. Learning to Predict Case Markers in Japanese. In **COLING-ACL**, pp. 1049–1056, 2006.

[18] 今村賢治, 齋藤邦子, 貞光九月, 西川仁. 小規模誤りデータからの日本語学習者作文の助詞誤り訂正. 自然言語処理, Vol. 19, No. 5, pp. 381–400, 2012.

[19] 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.

[20] Youichiro Ogawa and Kazuhide Yamamoto. Japanese Particle Error Correction employing Classification Model. In **IALP**, pp. 23–28, 2019.

[21] Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. Construction of an Evaluation Corpus for Grammatical Error Correction for Learners of Japanese as a Second Language. In **LREC**, pp. 204–211, 2020.

[22] 本間広樹, 小町守. 高速な文法誤り訂正機能を持つ日本語ライティング支援システムの構築. 人工知能学会論文誌, Vol. 37, No. 1, pp. 1–14, 2022.

[23] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In **BEA**, pp. 22–31, 2013.

[24] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 Shared Task on Grammatical Error Correction. In **CoNLL**, pp. 1–14, 2014.

[25] Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality. **TACL**, Vol. 4, pp. 169–182, 2016.

[26] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In **EACL**, pp. 229–234, 2017.

[27] Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. Predicting Grammaticality on an Ordinal Scale. In **ACL**, pp. 174–180, 2014.

[28] Jie He, Bo Peng, Yi Liao, Qun Liu, and Deyi Xiong. TGEA: An Error-Annotated Dataset and Benchmark Tasks for Text Generation from Pretrained Language Models. In **ACL-IJCNLP**, pp. 6012–6025, 2021.

[29] 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. 言語処理学会年次大会, pp. 1336–1339, 2020.

[30] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In **EMNLP**, pp. 66–71, 2018.

[31] Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In **NAACL**, pp. 568–572, 2012.

[32] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground Truth for Grammatical Error Correction Metrics. In **ACL-IJCNLP**, pp. 588–593, 2015.

[33] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In **ACL**, pp. 177–180, 2007.

[34] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. In **ICML**, pp. 1243–1252, 2017.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In **NeurIPS**, pp. 5998–6008, 2017.

[36] Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. **CL**, Vol. 29, No. 1, pp. 19–51, 2003.

[37] Kenneth Heafield. KenLM: Faster and Smaller Language Model Queries. In **WMT**, pp. 187–197, 2011.

[38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In **ACL**, pp. 311–318, 2002.

[39] Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In **ACL**, pp. 160–167, July 2003.

[40] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In **NAACL**, pp. 48–53, 2019.

[41] Shamil Chollampatt and Hwee Tou Ng. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In **AAAI**, pp. 5755–5762, 2018.

[42] Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. Massive Exploration of Pseudo Data for Grammatical Error Correction. **TASLP**, Vol. 28, pp. 2134–2145, 2020.

A　訂正者間の文単位の類似度

BLEU [38] を用いて訂正者間の文単位の類似度を測定する．BLEU は出力文と参照文中の N-gram の一致率をベースとした評価尺度であり，スコアが高ければ高いほど出力文と参照文の表層が一致していることを表す．表 5 に各訂正者間の BLEU スコアを示す．訂正文の分かち書きには Janome[16] を使用した．表 5 より，訂正者 D と訂正者 E の間のスコアが最も高いことが分かる．このことから，訂正者 D と訂正者 E は同じ表現を用いて訂正することが多いと考えられる．また訂正者 C は，他の訂正者とのスコアが低いことから，他の訂正者とは異なる表現で訂正することが多いと考えられる．

表 5: 各訂正者間の BLEU スコア．

出力文＼参照文	A	B	C	D	E
A	-	63.0	56.4	65.4	68.2
B	62.9	-	58.5	65.1	70.1
C	56.0	58.2	-	58.2	60.9
D	65.3	65.1	58.5	-	**72.3**
E	68.2	70.2	61.3	**72.5**	-

B　実験に使用したデータセット

データセット	原文数	参照文数	用途	評価器
Lang-8 コーパス [19]	970,273	1	訓練	-
Ours (dev)	515	5	検証	-
TEC-JL [21]	1,874	2	評価	M^2 scorer [31]
Ours (test)	1,030	5	評価	GLEU [32]

C　全ての参照文で評価した時の各文法誤り訂正モデルの性能

	TEC-JL			Ours
文法誤り訂正モデル	Prec.	Rec.	$F_{0.5}$	GLEU
訂正なし	-	-	-	52.5
SMT	**60.2**	9.8	29.6	53.5
CNN	53.1	19.1	39.2	55.1
Transformer	42.7	**32.4**	**40.1**	**56.2**

D　各文法誤り訂正モデルの出力例

学習者文	だから、友達は歩きにくいですから、もう今朝心斎橋や扇町へ行かなかったんです。
訂正者 A	友達が歩けないことを考慮して、今朝は心斎橋や扇町へ行かなかったです。
訂正者 B	友達が怪我のせいで歩きにくいので、今日は心斎橋や扇町へ遊びに行きませんでした。
訂正者 C	友達は歩きにくかったので、今朝は心斎橋や扇町へ行けませんでした。
訂正者 D	そのため、友達が歩きにくいので、心斎橋や扇町へ行くことは無くなりました。
訂正者 E	だから、友達は歩くのが大変で、今朝心斎橋や扇町へ行けなかったんです。
SMT	だから、友達は歩きにくいですから、もう今朝心斎橋や扇町へ行かなかったんです。
CNN	だから、友達は歩きにくいですから、もう今朝心斎橋や扇町へ行かなかったんです。
Transformer	だから、友達は歩きにくかったので、もう今朝心斎橋や扇町へ行かなかったんです。

16）https://github.com/mocobeta/janome