IMPARA-GED：言語モデルの文法誤り検出能力に着目した文法誤り訂正の参照文なし自動評価

坂井 優介　五藤 巧　渡辺 太郎
奈良先端科学技術大学院大学
{sakai.yusuke.sr9, goto.takumi.gv7, taro}@is.naist.jp

概要

本稿では，参照文なし文法誤り訂正自動評価手法である IMPARA-GED を提案する．我々は文法誤り訂正の自動評価手法である IMPARA に使用される品質推定モデルに着目し，文法誤り検出能力を強化した事前学習済み言語モデルを用いて，IMPARA-GED の品質推定モデルを構築した．文法誤り訂正自動評価手法のメタ評価用データセットである SEEDA を用いた性能評価実験の結果，IMPARA-GED は特に文単位の人手評価結果と最も高い相関を示す自動評価手法であることが示された．

1　はじめに

文法誤り訂正（Grammatical Error Correction; GEC）システムは，文法誤りや表記誤りを含む文（誤文）を受け取り，訂正後の文（訂正文）を出力する．これら GEC システムの自動評価は，参照文を用いた評価手法 [1, 2, 3] と参照文を用いない評価手法 [4, 5, 6] に大別でき，特に参照文を用いない評価手法には事前学習済み言語モデル（Pre-trained Language Model; PLM）がよく用いられる．例えば Scribendi Score [5] は，訂正文の評価値算出に PLM のパープレキシティを用いており，SOME [4] では流暢性・文法性・意味保存性のそれぞれ観点について PLM を追加学習することで評価器を構築する．また，IMPARA [6] は誤文と訂正文の類似度評価と訂正文の品質推定評価を組み合わせた評価法であり，その両方に PLM が用いられる．特に IMPARA は，誤文と訂正文のパラレルデータのみから品質スコア付きの擬似的な訂正文を自動生成することで，人手評価結果を必要とせずに品質推定モデルを学習可能な利点がある．

しかし，従来の IMPARA は品質推定モデルの学習に素の PLM を用いるが，PLM の事前学習から獲得される文法誤りに関する知識のみでは，文法誤りの情報を正確に捉えられるとは限らない．従って，IMPARA の学習のための擬似データ生成や，品質推定器の学習に改良の余地があると考えられる．

本稿では，参照文なし GEC 自動評価手法である IMPARA-GED を提案する．GEC システム構築において文法誤り検出（Grammatical Error Detection; GED）タスクによる追加学習の有用性を示した研究 [7, 8] から着想を得て，IMPARA-GED は PLM を GED タスクで追加学習後，IMPARA の品質推定モデル構築法を適用して作成を行う．我々は，まず IMPARA の類似度評価モデルに関する検証結果から，素の PLM が文法誤りに伴う類似度の変化を適切に捉えられない場合を示す．この事例より IMPARA において素の PLM の使用が不適切だと主張し，また評価尺度の定義に類似度評価を含めることの是非について議論する．その後，GED タスクでの追加学習後，IMPARA の品質推定モデル構築法を適用することで，IMPARA-GED を構築する．メタ評価ベンチマークの SEEDA [9] において IMPARA-GED は文単位の人手評価との相関が最も高いことがわかった．

2　IMPARA

IMPARA は訂正文の品質を評価する**品質推定モデル**と，誤文と訂正文の意味保存性を評価する**類似度評価モデル**から構成される．また類似度評価モデルは素の PLM を用いるが，品質推定モデルは独自の方法で PLM を追加学習することで構築される．

品質推定モデルの構築　品質推定モデルは，品質が高い訂正文 S_+ と品質が低い訂正文 S_- のペア (S_+, S_-) を入力として，品質の優劣の順序関係を学習することで構築される．これらの品質が異なる訂正文は，誤文と正文で構成される既存のパラレルデータから自動生成することができる．具体的には，誤文を正文に訂正するための編集集合を抽出し，各編集の影響度を，正文から編集を除いた時の

意味変化の度合いとして定量化する．また，編集集合から 2 つの異なる部分集合をサンプルし，それらを誤文に適用することで 2 つの異なる訂正文を得る．この時，すでに編集単位で影響度を計算しているため，影響度の総和をそれぞれの部分集合について計算することができる．この影響度の総和を訂正文の品質とみなして，品質が高い訂正文を S_+，低い訂正文を S_- として学習データ $\mathcal{T}$ とする．その後，学習データ $\mathcal{T}$ を用いて式（1）に示す損失関数 $\mathcal{L}^{\mathrm{QE}}$ を最小化するように品質推定モデル R を学習する．

$$\mathcal{L}^{\mathrm{QE}} = \frac{1}{|\mathcal{T}|} \sum_{(S_+,S_-)\in\mathcal{T}} \sigma(R(S_-) - R(S_+)). \quad (1)$$

ここで $\sigma(\cdot)$ はシグモイド関数である．また品質推定モデル R は誤文の埋め込み表現のうち，最終層の先頭トークンの表現を実数値へ線形変換している．

推論時のスコア算出方法　IMPARA は GEC システムへの誤文 I と，システムが出力した訂正文 O のペアに対して，品質推定モデル R と類似度評価モデル $\mathrm{sim}(\cdot)$ を用いることで，式（2）のように評価スコア $\mathrm{score}(I,O) \in [0,1]$ を算出する．

$$\mathrm{score}(I,O) = \begin{cases} \sigma(R(O)) & (\text{if } \mathrm{sim}(I,O;\mathrm{PLM}) > \theta) \\ 0 & (\text{otherwise}) \end{cases} \quad (2)$$

式（2）は類似度評価モデルの出力に閾値 θ を設定しており，誤文と訂正文の意味が大きく乖離するような事例について，品質推定モデルが不当に高い評価スコアの付与を防ぐフィルタの役割がある．

3　提案手法：IMPARA-GED

我々は IMPARA における類似度評価モデルの除去と，品質推定モデル構築時に GED タスクによる追加学習を行うことで，IMPARA-GED を構築する．

3.1　類似度評価モデルの除去

IMPARA は類似度評価モデルの出力に閾値を設けることで不当な訂正文をフィルタするが，類似度の評価性能は素の PLM に依存する．表 1 に SEEDA [9]（§4 で詳述）を用いたメタ評価において，様々な PLM を類似度評価モデルとして使用したときの結果を示す．表 1 より，PLM の種類によって結果が変動し，その多くで品質推定モデルのみの場合よりも相関は維持もしくは悪化する傾向にある．

この原因として，素の PLM を用いた類似度評価が誤ったフィルタリングを引き起こしていることが挙げられる．SEEDA 中の事例：誤文 “I think the family will stay mentally healty as it is , without having emtional stress .” と訂正文 “I think the family will stay mentally healthy without having emotional stress .” のペアにおいて，BERT-Large-uncased では類似度を 0.787 と計算する．ここで訂正文は誤文に含まれる誤りを訂正した文なので，本来類似度評価によってフィルタする必要はないが，IMPARA の標準設定値 0.9 ではフィルタされてしまう．この事例では ‘healthy’ の訂正を除くことで類似度は 0.926 まで上昇したため，BERT-Large-uncased の表記誤りに対する文意の理解の乏しさが原因であると考えられる．対照的に，フィルタすべき訂正文を受け入れる事例も存在する．BERT-Base において擬似的な誤文 “I like cats .” に対して意味を否定した訂正文 “I dislike cats .” は 0.980 の類似度が計算される．GEC では意味を否定する訂正は妥当ではないと考えられるためこの訂正文はフィルタされるべきだが，類似度の高さから受け入れられてしまう．これらの現象は，素の PLM が表層の微細な変化から文意の変化を測定する能力に乏しく，類似度評価が期待される役割とは異なる働きをしていることに起因するものと考えられる．

また，表 1 で類似度評価モデルを用いても相関が変化しない場合も散見されることから，基本的には誤文と乖離した訂正文が評価の対象となることは非常に稀であると仮定してよい．さらに，悪意のある

表 1: 類似度評価モデルに採用する PLM の違いによる評価性能への影響．評価用データとして SEEDA を使用．訂正スコアの評価にはチューニング済みの公開モデルを用いた．BART-Base を選択した場合，IMPARA と等価である．閾値 θ は IMPARA 同様 0.9 に固定．† が付与されている PLM は SEEDA 内の全ての事例に対して類似性スコアが閾値以上だった．

	システムレベル				文レベル			
	文単位		編集単位		文単位		編集単位	
GEC 評価手法	r	ρ	r	ρ	Acc.	τ	Acc.	τ
品質推定スコアのみ	.916	.902	.902	.965	.753	.506	.752	.504
†BERT-Base (original)	.916	.902	.902	.965	.753	.506	.752	.504
BERT-Large	.889	.867	.909	.916	.731	.463	.737	.474
BERT-Base-uncased	**.922**	.909	.903	.944	.746	.493	.745	.491
BERT-Large-uncased	.902	.895	.904	.951	.738	.476	.743	.487
ELECTRA-Base	.920	.902	.904	.965	.752	.505	.751	.503
†ELECTRA-Large	.916	.902	.902	.965	.753	.506	.752	.504
DeBERTa-v3-Base	.906	.916	.891	.958	.750	.500	.749	.498
DeBERTa-v3-Large	.915	.916	.900	.958	.749	.498	.749	.499
†ModernBERT-Base	.916	.902	.902	.965	.753	.506	.752	.504
ModernBERT-Large	.917	.903	.903	.965	.753	.505	.752	.503

訂正文が入力されうる問題は IMPARA だけでなく，SOME などの他の尺度にも見られる普遍的な問題である [5]．このことから，この点への対処は別軸で研究されることが望ましく，現状の SEEDA などに基づく自動評価のメタ評価では品質推定のみに注目した評価を実施すべきであると考える．以上の議論を踏まえ，提案する IMPARA-GED では類似度評価は除くこととし，式（3）により評価値を計算する．

$$score(I, O) = \sigma(R(O)) \tag{3}$$

3.2 GED タスクによる PLM の追加学習

IMPARA の品質推定モデルの構築に使用する訂正文のペアは，修正箇所に基づく影響度を基に作成される．しかし，3.1 節で述べたように素の PLM は誤りを十分に捉えきれてないと考えられる．そこで，提案法では GED タスクで追加学習を行うことで誤り情報をより正確に捉えたモデルを構築し，その後 IMPARA の学習法を適用することで品質推定モデルを構築する．GED モデルは Yuan ら [7] に従いトークン単位で誤りを検出するモデルとし，誤りラベルは 2 値，4 値，25 値，55 値をそれぞれ試行する．ここで 2 値は正解か誤りかのラベル，4 値は正解，挿入，削除，置換の編集操作に関するラベル，25 値は ERRANT [1] で定義される品詞などに基づくラベル，55 値はこれらを組み合わせたラベルである．これらのラベルに基づくトークン単位の教師データは，誤文と正文からなる既存のパラレルデータと，ERRANT によるそれらのアラインメントの結果から自動的に付与される．形式的には，N トークンで構成される誤文 $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ とその誤りラベル $\boldsymbol{t} = [t_1, t_2, \ldots, t_N]$ を用いて，次の目的関数 $\mathcal{L}^{\mathrm{GED}}$ を最小化することにより学習する．

$$\mathcal{L}^{\mathrm{GED}}(\boldsymbol{x}, \boldsymbol{t}) = -\frac{1}{N}\sum_{i=1}^{N} \log p(t_i | x_i, \boldsymbol{x}). \tag{4}$$

学習済みの GED モデルを用いて式（1）に従って品質推定モデルのための追加学習を行い，式（3）によって推論が実施される．このとき，トークン単位の誤り検出情報をより有効に活用するために，式（2）で先頭トークンの表現を用いる代わりに全てのトークン表現に対する平均プーリングを用いる．

4 実験設定

IMPARA-GED の構築 CoNLL-2013 [10] と FCE [11] をモデル構築用データセットとして使用した．CoNLL2013 は元の学習データを 8:1:1 の割合でそれぞれ学習用，開発用，テストデータに分割し，FCE はあらかじめ設定された分割のまま使用する．はじめに，GED モデルを Yuan ら [7] の設定に基づいて構築する．PLM を FCE と CoNLL2013 を合わせた学習データで 5 エポック学習し，FCE の開発用データで最良のスコアを達成するチェックポイントを最終的な GED モデルとする．次に作成した GED モデルを用いて，2 節の手順で品質推定モデルを構築する．Maeda ら [6] の設定に基づき，CoNLL-2013 の学習データから品質推定モデルの学習データを作成する．式（1）に従い GED モデルを 10 エポック学習し，CoNLL-2013 の開発用データでの結果が最良となるチェックポイントを選択する．これらを異なる 5 つのシード値で品質推定モデルを学習し，CoNLL2013 のテストデータで最良のモデルを最終的な品質推定モデルとした．なお，実験結果は GED のラベルの粒度の種類：2 値，4 値，25 値，55 値と，PLM の種類：BERT-Base [12]，BERT-Large [12]，DeBERTa-v3-Large [13]，ModernBERT-Large [14] のそれぞれの組み合わせについて報告する．

評価方法 SEEDA [9] によるメタ評価を実施する．SEEDA の TrueSkill に基づく人手評価について，文単位と編集単位の両方について相関を計算することとし，システム集合は Base 設定に従う．また，システム単位評価の相関係数として Pearson の積率相関係数 r と Spearman の順位相関係数 ρ，および文単位評価の相関係数として Accuracy（Acc.）と Kendall の順位相関係数 τ を報告する．ベースラインには，参照文を用いる評価方法として ERRANT [15, 1]，PT-ERRANT [16]，GREEN [3]，GLEU [17]，参照文を用いない評価方法として Scribendi Score [5]，SOME [4]，オリジナルの IMPARA [6] を報告する．また．大規模言語モデルによる GEC 評価システムとして，GPT-4-S [18] とその派生システムについても報告する．評価実験では gec-metrics[1] の実装と各システムの標準設定を使用するが，GPT-4-S については，文献 [18] の報告値を引用する．なお，システム評価については五藤ら [19] による報告結果を受け，全て TrueSkill [20] で評価結果を計算する．

5 実験結果と考察

表 2 に各 GEC 自動評価手法の SEEDA での評価結果を示す．表 2 の結果から，文単位に基づく評価で

1） https://github.com/gotutiyan/gec-metrics

表 2: SEEDA によるメタ評価結果．IMPARA-GED については学習に使用した PLM 名で示している．IMPARA-GED の各先頭要素は GED による追加学習を行わない場合の結果である．

	システムレベル				文レベル			
	文単位		編集単位		文単位		編集単位	
GEC 評価手法	r	ρ	r	ρ	Acc.	τ	Acc.	τ
ERRANT	.763	.706	.881	.895	.594	.189	.608	.217
PT-ERRANT	.870	.797	.924	.951	.582	.165	.592	.184
GREEN	.855	.846	.912	.965	.600	.199	.574	.148
GLEU	.863	.846	.909	.965	.672	.343	.673	.347
Scribendi	.674	.762	.837	.888	.660	.320	.672	.345
SOME	.932	.881	.893	.944	.778	.555	.766	.532
IMPARA	.939	.923	.901	.944	.753	.506	.752	.504
GPT-4-S	.887	.860	.960	.958	.784	.567	.798	.595
+文法性	.888	.867	.961	.937	.796	.592	.807	.615
+流暢性	.913	.874	**.974**	**.979**	.796	.592	.807	.615
+意味保持性	.958	.881	.911	.960	.810	.620	**.813**	**.626**
BERT-Base	.915	.895	.875	.930	.756	.512	.754	.508
+2 値	.916	.909	.850	.902	.773	.545	.763	.527
+4 値	.908	.902	.859	.923	.787	.574	.774	.548
+25 値	.925	.902	.875	.923	.771	.543	.752	.503
+55 値	.900	.902	.842	.923	.763	.526	.750	.499
BERT-Large	.937	.909	.898	.937	.783	.568	.765	.530
+2 値	.949	.923	.906	.944	.764	.529	.755	.510
+4 値	.827	.825	.767	.867	.701	.403	.686	.371
+25 値	.934	.902	.886	.923	.782	.564	.762	.524
+55 値	.915	.895	.879	.930	.767	.534	.758	.517
DeBERTa-v3	.960	.937	.912	.944	.784	.568	.779	.558
+2 値	.951	.923	.895	.916	.797	.593	.784	.568
+4 値	.939	.895	.899	.916	.793	.585	.772	.544
+25 値	.945	.930	.906	.930	.801	.602	.786	.573
+55 値	.955	.930	.913	.958	.782	.564	.763	.527
ModernBERT	.949	.909	.912	.937	.767	.533	.749	.497
+2 値	.971	.930	.919	.930	**.829**	**.658**	.797	.594
+4 値	.964	.916	.926	.923	.812	.624	.794	.588
+25 値	**.972**	.937	.933	.944	.801	.603	.783	.567
+55 値	.965	**.951**	.910	.909	.749	.498	.741	.483

は，GED による追加学習を行うことで評価結果が改善する傾向が観察できる．特に ModernBERT-Large を使用した場合，2 値分類による GED タスクで事前学習を行うことにより，既存手法と比較して最も高い評価結果が達成できた．GED の学習については，誤り分類クラス数を増やせばよいわけではなく，2 値分類でも十分性能が向上していることがわかる．さらに，編集に基づく人手評価ではシステムレベルと文レベル両方で GED の学習効果が十分に発揮されなかったが，文単位に基づく人手評価については GED の学習効果がよく現れている．IMPARA は文

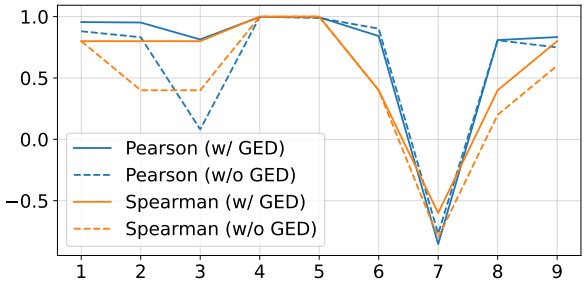

図 1: window を 4 とした window-analysis の結果．横軸は人手順位の開始順位であり，例えば $x = 2$ は人手評価の 2 位から 6 位までのシステムを対象とした結果を示す．ModernBERT-Large の GED を行わなかったモデルと 2 値ラベルによる GED で追加学習を行ったモデルで比較している．

単位で評価するためこの結果は自然であり，より正確な文単位評価のために GED の学習が有効であった．また，通常の IMPARA の学習において，BERT 以外の言語モデルにしたとき，必ずしも同等の結果が得られるとは限らず，BERT-Base から BERT-Large に変更しても，評価結果が向上するとは限らなかった．これは 3.1 節の議論と同様に，PLM は誤りを捉えきることができていないことが要因として挙げられる．そのため，IMPARA に他の PLM を用いることは検討の余地がある一方で，ModernBERT のような PLM では，GED による事前学習を行うことで大幅な誤り認識能力の獲得を示すモデルもあるため，追加学習として GED タスクを行うことには一定の優位性があると結論付けられる．

また図 1 に ModernBERT-Large における文単位の評価に対する window-analysis の結果を示す．図 1 より，GED の追加学習は上位のシステムの評価性能向上に寄与していることが伺える．

6 まとめ

本稿では，PLM の文法誤り検出能力に着目し，GED タスクで PLM の追加学習を行うことで，参照文なし文法誤り訂正自動評価手法である IMPARA-GED を提案した．ModernBERT を PLM として用いたとき，2 値ラベルに基づいた GED タスクにより追加学習を行った結果，SEEDA において，既存手法と比較して人手評価結果と最も高い相関を示した．さらに，window-analysis の結果，システム評価において，特に上位システムの評価性能が向上していることがわかった．

参考文献

[1] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In Regina Barzilay and Min-Yen Kan, editors, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[2] Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. Taking the correction difficulty into account in grammatical error correction evaluation. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, pp. 2085–2095, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[3] Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. n-gram F-score for evaluating grammatical error correction. In Saad Mahamood, Nguyen Le Minh, and Daphne Ippolito, editors, Proceedings of the 17th International Natural Language Generation Conference, pp. 303–313, Tokyo, Japan, September 2024. Association for Computational Linguistics.

[4] Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, Proceedings of the 28th International Conference on Computational Linguistics, pp. 6516–6522, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[5] Md Asadul Islam and Enrico Magnani. Is this the end of the gold standard? a straightforward reference-less grammatical error correction metric. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 3009–3015, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[6] Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. IMPARA: Impact-based metric for GEC using parallel data. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, Proceedings of the 29th International Conference on Computational Linguistics, pp. 3578–3588, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[7] Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. Multi-class grammatical error detection for correction: A tale of two systems. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 8722–8736, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] 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 Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4248–4254, Online, July 2020. Association for Computational Linguistics.

[9] Masamune Kobayashi, Masato Mita, and Mamoru Komachi. Revisiting meta-evaluation for grammatical error correction. Transactions of the Association for Computational Linguistics, Vol. 12, pp. 837–855, 2024.

[10] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In Hwee Tou Ng, Joel Tetreault, Siew Mei Wu, Yuanbin Wu, and Christian Hadiwinoto, editors, Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–12, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[11] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 180–189, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[13] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In The Eleventh International Conference on Learning Representations, 2023.

[14] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.

[15] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In Yuji Matsumoto and Rashmi Prasad, editors, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 825–835, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[16] Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. Revisiting grammatical error correction evaluation and beyond. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 6891–6902, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[17] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In Chengqing Zong and Michael Strube, editors, Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 588–593, Beijing, China, July 2015. Association for Computational Linguistics.

[18] Masamune Kobayashi, Masato Mita, and Mamoru Komachi. Large language models are state-of-the-art evaluator for grammatical error correction. In Ekaterina Kochmar, Marie Bexte, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Anaïs Tack, Victoria Yaneva, and Zheng Yuan, editors, Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024), pp. 68–77, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[19] 五藤巧, 坂井優介, 渡辺太郎. 文法誤り訂正における人手評価と自動評価の乖離とその解決. 言語処理学会第 31 回年次大会発表論文集, March 2025. (To Appear).

[20] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill™: A bayesian skill rating system. In Advances in Neural Information Processing Systems, Vol. 19. MIT Press, 2006.

A　本稿で使用した PLM の詳細

表 3 に本稿で使用した PLM と HuggingFace ID の対応を示す.

表 3: 本稿で言及した PLM と HuggingFace ID の対応

本稿でのモデル名	HuggingFace ID
BERT-Base	google-bert/bert-base-cased
BERT-Large	google-bert/bert-large-cased
BERT-Base-uncased	google-bert/bert-base-uncased
BERT-Large-uncased	google-bert/bert-large-uncased
ELECTRA-Base	google/electra-base-discriminator
ELECTRA-Large	google/electra-large-discriminator
DeBERTa-v3-Base	microsoft/deberta-v3-large
DeBERTa-v3-Large	microsoft/deberta-v3-large
ModernBERT-Base	answerdotai/ModernBERT-base
ModernBERT-Large	answerdotai/ModernBERT-large

B　実験に使用したツール

IMPARA-GED の学習設定については GED モデル構築に関しては Yuan ら [7]、品質評価モデル構築には Maeda らの IMPARA [6] の設定を使用した．GED の学習には，ged_baselines（https://github.com/gotutiyan/ged_baselines）を用いた．また，IMPARA の学習には再現実装（https://github.com/gotutiyan/IMPARA）を使用した．表 1 における品質評価モデルは HuggingFace 上の再現実装モデルである IMPARA-QE（https://huggingface.co/gotutiyan/IMPARA-QE）を用いた．なおハイパーパラメタなどの設定値は，言及がない限りこれら使用ツールのデフォルト値を用いた．