大規模言語モデルによる日本語文法誤り訂正の性能評価

花房 健太郎 [1]　宮田 莉奈 [1]　梶原 智之 [1,2]　北岡 佑一 [3]　荒木 亮 [3]　真嘉比 愛 [3]
[1] 愛媛大学大学院理工学研究科　[2] 大阪大学 D3 センター　[3] ちゅらデータ株式会社
{hanafusa@ai.,miyata@ai.,kajiwara@}cs.ehime-u.ac.jp
{y.kitaoka, m.araki, a.makabi}@churadata.okinawa

概要

本研究では，日本語における大規模言語モデルの文法誤り訂正の性能を広く調査する．英語において大規模言語モデルを用いた文法誤り訂正の研究が進んでいる一方，日本語ではモデル間の性能差や誤り傾向に関する体系的な分析は充分になされていない．本研究では，日本語特化モデルを含む 15 種類の大規模言語モデルを対象に，0-shot, 10-shot, Fine-tuning の設定で文法誤り訂正の性能を評価した．実験の結果，日本語特化モデルが多言語モデルよりも高い訂正性能を示すことが明らかになった．

1　はじめに

大規模言語モデル (Large Language Model: LLM) は，文脈内学習 [1] や指示チューニング [2] などの手法の進展により，様々な自然言語処理タスクで高い性能を示している．日本語においても，日本語を中心とする大規模コーパスで事前学習された LLM-jp [3] や，英語モデルに対する継続事前学習によって日本語能力を強化した Swallow [4] などが開発され，言語生成性能が大きく向上している．これらの LLM は，機械翻訳や要約をはじめとする多様なタスクにおいて評価が行われてきた [3–5].

文法誤り訂正は，所与のテキストに含まれる文法的な誤りを自動的に検出・修正するタスクであり，言語教育・文章校正支援・入力補助など幅広い応用を有する．英語においては，大規模な誤り訂正コーパスの整備やモデル比較を通じて，LLM を含む手法の性能や訂正傾向が詳細に分析されてきた [6–8]. 一方で，日本語の文法誤り訂正に関しては，LLM の性能の体系的な分析は充分になされていない．特に，誤りの種類ごとの訂正傾向や，日本語特有の語順や活用などの誤りに対する LLM の訂正能力は明らかになっていない．

本研究では，日本語における LLM の文法誤り訂正の能力を明らかにすることを目的とし，既存の日本語文法誤り訂正コーパスを用いて，複数の LLM の性能を体系的に評価する．評価実験の結果，Few-shot 文脈内学習と比較して，Fine-tuning が文法誤り訂正の性能を大幅に改善することを確認した．また，日本語に特化した LLM は，多言語モデルと比べて，助詞や活用の誤りに対して高い訂正能力を示すことがわかった．一方で，一部のモデルでは，文意を変えてしまう過剰訂正も見られ，LLM による日本語文法誤り訂正の課題も明らかとなった．

2　関連研究

2.1　近年の文法誤り訂正

英語の文法誤り訂正においては，LLM を用いた手法の開発やその性能に関する詳細な分析が進められている．具体的には，LLM の訂正傾向を分析した研究 [6] や，過剰訂正を抑制する学習手法に関する研究 [7, 8] などが報告されている．

日本語の文法誤り訂正においては，小山ら [9] が，BART [10] や T5 [11] などの比較的小規模なモデルを対象とする比較によって，モデルごとに流暢性や意味保存性の傾向が異なることを示している．しかし，日本語の文法誤り訂正において，LLM の規模や学習手法の違いが訂正性能に与える影響は依然として明らかになっていない．本研究では，複数の LLM を対象とする日本語文法誤り訂正の性能評価によって，その特性を明らかにする．

2.2　日本語文法誤り訂正データセット

日本語における文法誤り訂正の訓練および評価には，主に以下のデータセットが用いられている．

- **日本語 Wikipedia 入力誤りデータセット** [12]: Wikipedia の編集履歴から構築された大規模な

データセットである．母語話者が執筆時に発生させる誤字・脱字・衍字・転字・かな漢字変換に伴う誤変換を主な訂正対象としている．

- **Lang-8** [13]: 語学学習 SNS「Lang-8」の添削ログから構築された多言語学習者コーパスである．日本語学習者文が含まれており，日本語学習者に特有の文法誤りを多く含む点が特徴である．
- **TEC-JL** [9]: Lang-8 の一部を再添削して構築された評価用コーパスであり，「最小限の訂正」を重視している．助詞や助動詞の選択，品詞ごとの誤りのラベル（誤用タグ）が付与されており，品詞別の詳細な誤り分析が可能である．
- **FLUTEC** [14]: TEC-JL と同様に Lang-8 の一部を再添削して構築された評価用コーパスであり，文法的な正しさに加えて，「流暢な訂正」を重視している点が特徴である．
- **NAIST 誤用コーパス** [15]: 日本語学習者の課題作文から構築されたコーパスであり，約 76 種類の誤用タグが付与されたコーパスである．誤用タイプは「文法的誤用」や「語彙的誤用」といった大分類から，助詞や語彙選択といった詳細な分類へと階層化されている．

本研究では，これらのデータセットを用いて 15 種類の LLM の文法誤り訂正性能を評価する．

3 評価実験

3.1 実験設定

データセット　訓練データには，Wikipedia の編集履歴から収集された日本語 Wikipedia 入力誤りデータセット (v2) (JWTD: Japanese Wikipedia Typo Dataset) [12] と日本語学習者文を収集した Lang-8 [13] を用いた．Lang-8 を用いて訓練したモデルは，小山ら [9] の設定に従い，検証には FLUTEC [14] を，評価には FLUTEC [14]・TEC-JL [9]・NAIST 誤用コーパス [15] を用いた．各データの内訳を表 1 に示す．

モデル　本実験では，以下に示す 15 種類の LLM を評価対象とした．なお，言語モデリングの事前学習に加えて指示チューニングをしたモデルが利用可能な場合は，事前学習のみのモデル (Base) と指示チューニングモデル (Instruct) の両方を評価した．

日本語データで事前学習　日本語データや英語，ソースコードなどを用いて事前学習したモデルとして，LLM-jp[1)2)3)4)] [3] を用いた．

英語モデルを継続事前学習　英語 LLM に対して日本語データを用いて継続事前学習した LLM として，Swallow[5)6)] [4] を用いた．

表 1　データセットの統計（文対数）

データセット	訓練用	検証用	評価用
JWTD	696,190	5,440	1,127
Lang-8	1,042,932	-	-
TEC-JL	-	-	1,702
FLUTEC	-	1,047	1,029
NAIST 誤用コーパス	-	-	6,587

多言語モデル　英語データを中心に複数言語のテキストを用いて事前学習された多言語モデルとして，Llama[7)8)] [16]・Qwen[9)10)11)12)13)] [17]・gpt-oss[14)]・GPT-5.2 Thinking（以降 GPT-5.2 と表記）[15)] を用いた．

評価方法　GPT-5.2 については，0-shot および 10-shot の Few-shot 文脈内学習による評価のみを行った．それ以外のモデルについては，Few-shot 文脈内学習に加えて，教師ありデータを用いた Fine-tuning による評価も実施した．

ハイパーパラメータ　学習率は $1e-4$, $2e-5$, $1e-5$, $2e-6$, $1e-6$ の中から検証用データセットにおいて最も高い性能を示したものを選択した．最適化手法に Adam [18] を使用し，7 ステップの early-stopping を適用した．バッチサイズは 16 に設定した．LLM の Fine-tuning には LoRA (Low-Rank Adaptation) [19] を使用し，ランクを $r = 16$，スケーリング係数を $\alpha = 16$，dropout 率を 0.05 に設定した．

評価　性能評価には，以下の自動評価指標を用いた．日本語 Wikipedia 入力誤りデータセットに対しては ERRANT (ERRor ANnotation Toolkit) [20] を用いて $F_{0.5}$ 値を算出した．また，TEC-JL および NAIST 誤用コーパスについては Max Match (M^2) [21] を用いて評価した．さらに，FLUTEC については GLEU$^+$ [22, 23] を用いて性能を評価した．

3.2 実験結果

本節では，データセットの特性，指示チューニングの有無，学習手法の違いという 3 つの観点から性能を分析する．図 1 にこれらの実験結果を示す．なお，詳細な実験結果は付録の表 4 に掲載する．

各データセットにおけるモデルの性能比較　図 1 (a) に，各データセットにおけるモデルの性能比較を示す．JWTD では，Fine-tuning を施したモデルが総じて高いスコアを示した．特に日本語 LLM の性

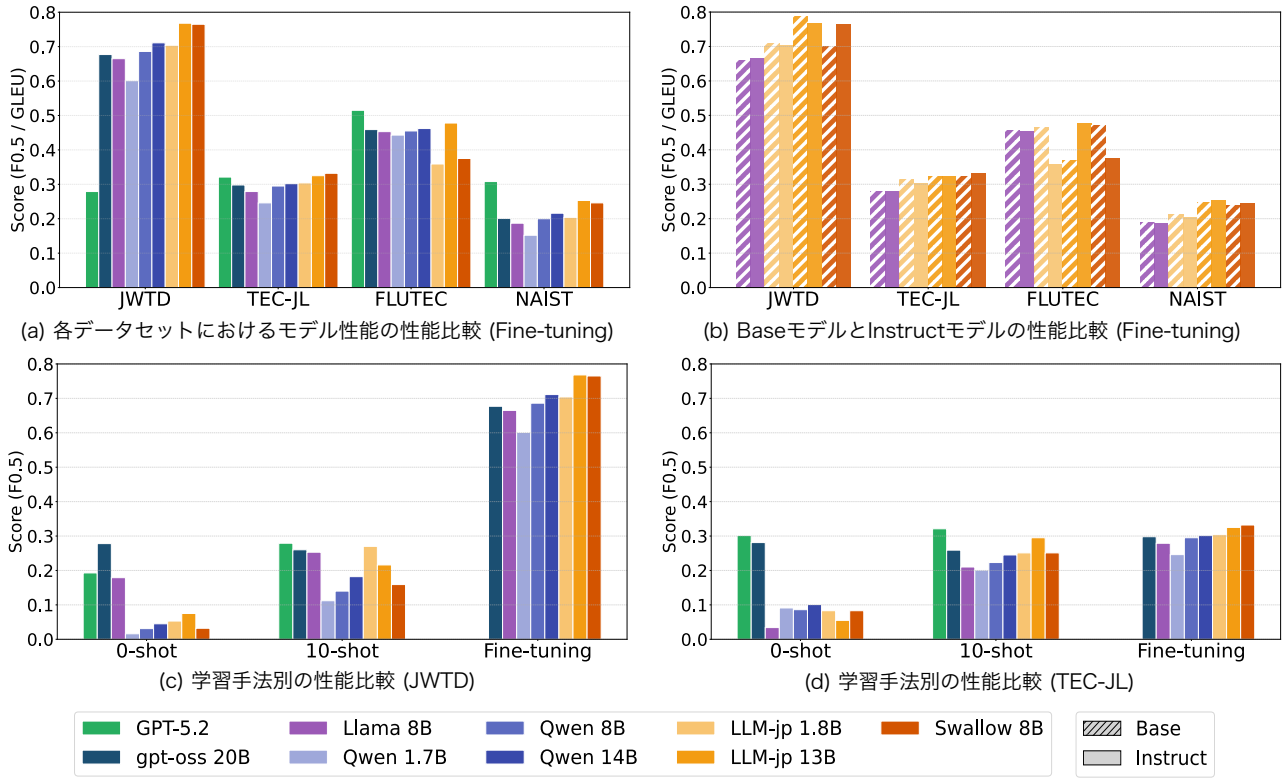

図 1　自動評価の結果

能が高く，小規模な LLM-jp 1.8B であっても Qwen 14B と同等の性能を達成している．これらの結果から，母語話者によるタイプミスを中心とした誤りは，LLM にとって比較的訂正しやすいことが示唆される．一方で，TEC-JL・FLUTEC・NAIST 誤用コーパスといった学習者コーパスでは，JWTD と比較して全体的にスコアが低下した．特に TEC-JL ではいずれのモデルにおいても性能が低く，LLM にとって最小訂正が難しいタスクであることが示された．FLUTEC では，日本語 LLM に加えて英語 LLM も高い性能を示しており，多くのモデルにおいて一定程度流暢な訂正が可能であることが確認された．また，FLUTEC および NAIST 誤用コーパスにおいては，Fine-tuning をしていない GPT-5.2 が，Fine-tuning 済みの他のモデルを上回る最も高い性能を示した．

Base モデルと Instruct モデルの比較　図 1 (b) に，Base モデルと Instruct モデルの性能比較を示す．多くのモデルにおいて Instruct モデルの方が高い性能を示す傾向が見られるものの，Base モデルも Instruct モデルと同程度の性能を達成している．特に，JWTD では LLM-jp 13B の Base モデルが最も高い性能を示している．これらの結果から，Base モデルと Instruct モデルの間に顕著な性能差は生じにくいことが示唆される．

Few-shot 文脈内学習と Fine-tuning の比較　図 1 (c) および (d) に，JWTD および TEC-JL における学習手法別の性能比較を示す．0-shot 設定に着目すると，GPT-5.2 や gpt-oss は，他のモデルと比較して高い性能を示した．一方で，両データセットにおいて，0-shot および 10-shot の Few-shot 文脈内学習と比較して，Fine-tuning をすることで性能が大幅に向上することが確認された．特に JWTD ではこの傾向が顕著であり，多くのモデルでスコアの大幅な改善が見られた．TEC-JL においても Fine-tuning の有効性は確認されたものの，JWTD ほどの性能向上が得られないモデルも存在し，学習者特有の多様な誤りに対する汎化の難しさが示唆される．

3.3　分析

誤りタイプ　表 2 の中央列に，TEC-JL における各モデルの誤りタイプ別 Recall を示す．GPT-5.2 はすべての誤りタイプにおいて最も高い性能を示した．GPT-5.2 を除くモデル間では，日本語 LLM が多言語モデルと比較して全体的に高い性能を示しており，特に助詞や助動詞の選択，動詞の活用に関する誤りにおいて顕著な差が確認された．一方で，いずれのモデルにおいても，助動詞の選択や名詞に関するスコアは相対的に低く，文脈理解を要する誤りの

表 2　TEC-JL における各モデルの誤りタイプ別の Recall (%) と一文あたりの平均推論時間 (s)

モデル	サイズ	誤りタイプ								推論時間
		助詞	その他	助動詞の選択	綴り	句読点	名詞	動詞の活用	動詞の選択	
LLM-jp	1.8B	56.06	38.36	48.37	45.11	**38.33**	33.33	43.94	61.29	0.91
LLM-jp	13.0B	**61.62**	**40.43**	**52.63**	52.17	34.43	**42.37**	56.25	**67.69**	2.58
Swallow	8.0B	60.96	40.16	50.32	**56.99**	37.29	38.98	**60.29**	65.08	1.38
Llama	8.0B	52.07	33.04	43.51	53.01	24.53	35.59	40.30	54.10	0.88
Qwen	1.7B	47.83	28.00	44.44	39.78	25.00	30.77	34.33	57.41	0.87
Qwen	8.0B	54.96	32.62	46.15	53.85	35.19	37.50	45.59	70.00	1.63
Qwen	14.0B	56.50	32.44	49.34	51.65	34.92	44.07	49.23	59.70	2.31
gpt-oss	20.0B	53.87	38.98	43.62	48.37	29.31	23.64	45.45	62.30	7.85
GPT-5.2	-	**77.19**	**59.71**	**54.43**	**79.70**	**71.43**	**59.15**	**67.14**	**76.00**	(0.87)

表 3　事例ごとの出力例（赤色：誤り，青色：正解）

日本語 LLM のみが訂正できた例			過剰訂正をした例
入力文		可愛そうな境遇だと言われ幼少期を過ごしてきた.	助言を受けたいです。
LLM-jp	1.8B	哀れな境遇だと言われ幼少期を過ごしてきた.	アドバイスを受けたいです。
LLM-jp	13.0B	可哀想な境遇だと言われ幼少期を過ごしてきた.	アドバイスを受けたいです。
Swallow	8.0B	可哀想な境遇だと言われ幼少期を過ごしてきた.	助言をいただきたいです。
Qwen	1.7B	可愛そうな境遇だと言われ幼少期を過ごした.	助言をしたいです。
Qwen	14.0B	可哀相な境遇だと言われ幼少期を過ごしてきた.	助言をいただきたいです。
Llama	8.0B	可哀そうな境遇だと言われ幼少期を過ごしてきた.	助言を求めたいです。
gpt-oss	20.0B	可哀相な境遇だと言われ幼少期を過ごしてきた.	助言をいただきたいです。
GPT-5.2	-	かわいそうな境遇だと言われ幼少期を過ごしてきた.	助言をいただきたいです。
正解文		可哀想な境遇だと言われ幼少期を過ごしてきた.	助言を受けたいです。

訂正は依然として難しいことが示唆される．

推論時間　本実験における推論時間の計測には，Intel Xeon Silver 4208 および NVIDIA RTX A6000 を搭載した計算機を使用した．表 2 の右列に，各モデルの推論時間を示す．推論時間は概ねモデルサイズと正の相関を示しており，特に gpt-oss は他のモデルと比べて著しく長い推論時間を要した．

日本語 LLM の訂正事例　表 3 の左側に，日本語 LLM のみが正しく訂正できた事例を示す．Swallow 8B や LLM-jp 13B は，漢字の誤用を正解文と一致する形で訂正している．一方で，小規模な LLM-jp 1.8B は類義語への置換が生じ，Qwen 1.7B では誤りを訂正できずに文末のみが変更されるなど，パラメタ数の少ないモデルでは正確な訂正が困難となる傾向が見られた．また，GPT-5.2 を除く英語 LLM では「可哀相」など誤った漢字表記を出力する例が確認された．一方，GPT-5.2 は「かわいそう」と出力しており，誤りの訂正自体は達成しているものの，正解文とは表記が一致しなかった．これは，Few-shot 文脈内学習において，データセット固有の表記規則を十分に学習できなかった可能性が考えられる．

過剰訂正の事例　訂正を必要としない箇所に対しても書き換えを行う過剰訂正の事例が，多くのモデルで見られた．表 3 の右側に，過剰訂正の具体例を示す．LLM-jp 系列のモデルは「アドバイス」への書き換えをして，他の多くのモデルでは「助言をいただきたいです。」のように，より丁寧な表現への訂正が生じた．特に Qwen 1.7B は「助言をしたいです。」と出力され，文意を反転させる重大な誤りが確認された．これらの結果から，流暢性の向上を優先する生成傾向が，不必要な修正や意味の変化を引き起こす要因となっていることが示唆される．

4　おわりに

本研究では，LLM による日本語の文法誤り訂正の性能を評価した．既存の日本語文法誤り訂正データセットを用いて 15 種類の LLM を比較した結果，日本語特化 LLM は，多言語 LLM と比べて，助詞や活用といった文法誤りに加え，漢字表記の訂正において高い性能を示した．しかし，助動詞や名詞の誤り訂正が難しいことや，文意を変化させる過剰訂正が生じるなどの課題も明らかとなった．

参考文献

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In **Proc. of NeurIPS**, pp. 1877–1901, 2020.

[2] 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.

[3] LLM-jp. LLM-jp: A Cross-organizational Project for the Research and Development of Fully Open Japanese LLMs. **arXiv:2407.03963**, 2024.

[4] Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. In **Proc. of CoLM**, 2024.

[5] 樽本空宙, 畠垣光希, 宮田莉奈, 梶原智之, 二宮崇. ChatGPT の日本語生成能力の評価. 自然言語処理, Vol. 31, No. 2, pp. 349–373, 2024.

[6] Anisia Katinskaia and Roman Yangarber. GPT-3.5 for Grammatical Error Correction. In **Proc. of LREC-COLING**, pp. 7831–7843, 2024.

[7] Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. Adapting LLMs for Minimal-Edit Grammatical Error Correction. In **Proc. of BEA**, pp. 118–128, 2025.

[8] Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. Edit-Wise Preference Optimization for Grammatical Error Correction. In **Proc. of COLING**, pp. 3401–3414, 2025.

[9] 小山碧海, 喜友名朝視顕, 小林賢治, 新井美桜, 三田雅人, 岡照晃, 小町守. 日本語文法誤り訂正のための誤用タグ付き評価コーパスの構築. 自然言語処理, Vol. 30, No. 2, pp. 330–371, 2023.

[10] 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.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. **JMLR**, Vol. 21, No. 140, pp. 1–67, 2020.

[12] 田中佑, 村脇有吾, 河原大輔, 黒橋禎夫. 日本語 Wikipedia の編集履歴に基づく入力誤りデータセットと訂正システムの構築. 自然言語処理, Vol. 28, No. 4, pp. 995–1033, 2021.

[13] 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. 自然言語処理, Vol. 28, No. 5, pp. 420–432, 2013.

[14] 木山朔, 上坂奏人, 佐藤郁子, 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, pp. 1704–1709, 2022.

[15] 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.

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models. **arXiv:2302.13971**, 2023.

[17] Qwen Team. Qwen3 Technical Report. **arXiv:2505.09388**, 2025.

[18] Diederik P. Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimization. In **Proc. of ICLR**, 2015.

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

[20] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In **Proc. of ACL**, pp. 793–805, 2017.

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

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

[23] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. GLEU Without Tuning. **arXiv:1605.02592**, 2016.

付録：詳細な実験結果

表 4 に，本研究で評価した全モデルの自動評価結果を示す．

表 4　全モデルの自動評価の結果（0-shot・Few-shot・Fine-tuning）

Model	Size	Instruct	method	Wikipedia	TEC-JL	FLUTEC	NAIST 誤用コーパス
				$F_{0.5}$	$F_{0.5}$	GLEU	$F_{0.5}$
LLM-jp	1.8B[1)]	-	Fine-tuning	0.710	0.315	0.465	0.213
	1.8B[2)]	✓	0-shot	0.053	0.083	0.041	0.055
			10-shot	0.270	0.251	0.386	0.057
			Fine-tuning	0.704	0.304	0.359	0.204
	13.0B[3)]	-	Fine-tuning	0.786	0.322	0.368	0.248
	13.0B[4)]	✓	0-shot	0.075	0.055	0.058	0.057
			10-shot	0.216	0.295	0.299	0.223
			Fine-tuning	0.768	0.325	0.478	0.253
Swallow	8.0B[5)]	-	Fine-tuning	0.699	0.322	0.470	0.237
	8.0B[6)]	✓	0-shot	0.032	0.083	0.036	0.095
			10-shot	0.159	0.251	0.238	0.245
			Fine-tuning	0.765	0.332	0.375	0.246
Llama	8.0B[7)]	-	Fine-tuning	0.659	0.279	0.455	0.188
	8.0B[8)]	✓	0-shot	0.179	0.034	0.145	0.140
			10-shot	0.253	0.210	0.431	0.177
			Fine-tuning	0.665	0.279	0.453	0.187
Qwen	0.6B[9)]	✓	0-shot	0.036	0.121	0.216	0.100
			10-shot	0.047	0.154	0.353	0.123
			Fine-tuning	0.571	0.201	0.429	0.113
	1.7B[10)]	✓	0-shot	0.016	0.091	0.039	0.087
			10-shot	0.112	0.201	0.269	0.159
			Fine-tuning	0.601	0.246	0.443	0.152
	4.0B[11)]	✓	0-shot	0.024	0.085	0.039	0.079
			10-shot	0.154	0.199	0.144	0.168
			Fine-tuning	0.664	0.275	0.451	0.182
	8.0B[12)]	✓	0-shot	0.031	0.086	0.033	0.096
			10-shot	0.140	0.223	0.208	0.199
			Fine-tuning	0.686	0.295	0.455	0.200
	14.0B[13)]	✓	0-shot	0.045	0.101	0.044	0.101
			10-shot	0.182	0.245	0.241	0.218
			Fine-tuning	0.711	0.302	0.462	0.216
gpt-oss	20.0B[14)]	✓	0-shot	0.278	0.281	0.456	0.216
			10-shot	0.260	0.259	0.434	0.228
			Fine-tuning	0.677	0.298	0.459	0.201
GPT-5.2[15)]	-	✓	0-shot	0.193	0.302	0.499	0.293
			10-shot	0.279	0.321	0.515	0.308

1）https://huggingface.co/llm-jp/llm-jp-3.1-1.8b
2）https://huggingface.co/llm-jp/llm-jp-3.1-1.8b-instruct4
3）https://huggingface.co/llm-jp/llm-jp-3.1-13b
4）https://huggingface.co/llm-jp/llm-jp-3.1-13b-instruct4
5）https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-v0.5
6）https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.5
7）https://huggingface.co/meta-llama/Llama-3.1-8B
8）https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
9）https://huggingface.co/Qwen/Qwen3-0.6B
10）https://huggingface.co/Qwen/Qwen3-1.7B
11）https://huggingface.co/Qwen/Qwen3-4B
12）https://huggingface.co/Qwen/Qwen3-8B
13）https://huggingface.co/Qwen/Qwen3-14B
14）https://huggingface.co/openai/gpt-oss-20b
15）https://platform.openai.com/docs/models/gpt-5.2
2025 年 12 月 26 日〜2025 年 12 月 28 日にアクセス