ehiMetrick: NLP2025 自動評価ハック Shared Task 文法誤り訂正部門

杉山 誠治 [1]　森岡 拓 [2]　高山 隼矢 [2]　梶原 智之 [2]
[1] 愛媛大学工学部　[2] 愛媛大学大学院理工学研究科
{sugiyama@ai., morioka@ai., takayama@ai., kajiwara@}cs.ehime-u.ac.jp

概要

本研究では，文法誤り訂正における自動評価の脆弱性について検証する．自動評価は，人手評価に比べて低コストに訂正文を定量評価できる．一方，訂正文として不適切な文に対しても不当に高い評価を与えてしまう可能性がある．文法誤り訂正の自動評価のうち，埋め込みベースの IMPARA では入力文と訂正文の類似度に基づくフィルタリング機構の脆弱性，大規模言語モデルを用いる GPT-4-S ではプロンプトへの指示追従性に起因する脆弱性を指摘し，それらに基づいて不当に高い評価を誘発する出力文の作成方法を提案する．評価実験では，提案手法に基づいて文法訂正モデルの出力文を改竄した場合に，品質が向上していないにも関わらずスコアが不当に増加することを確認し，実際に各評価指標に脆弱性が存在することを示した．

1　はじめに

文法誤り訂正 (GEC：Grammatical Error Correction) は，文法的に誤りを含む文を文法的に正しい文に編集するタスクであり，言語学習者を支援する技術である．文法誤り訂正の自動評価では，誤りを含む文を入力文として，システムが編集した出力文の質を定量的に評価する．自動評価の方法は，入力文と出力文の表層的な一致度に基づく評価 [1–4]，事前学習済みモデルを用いた埋め込みベースの評価 [5,6]，大規模言語モデル (LLM: Large Language Model) を用いた評価 [7] に大別できる．これらの自動評価は，人手評価よりも低コストに評価できる一方で，文法的に誤りを含んだ文に対しても高い評価を与えてしまうという脆弱性を持つ．先行研究 [8] では，2013 および 2014 CoNLL GEC Shared Task [9,10] で使用された MaxMatch (M^2) [11] について，入力文の語句編集において，適切な訂正文よりも不適切な訂正文に高い評価を与えることがあると指摘されている．

本研究では，このような文法誤り訂正の自動評価の脆弱性を見つけることが目的であり，埋め込みベースの評価である IMPARA [6] と LLM を用いた評価である GPT-4-S [7] の脆弱性について検証する．IMPARA においては入力文と出力文の類似度に基づくフィルタリング機構の脆弱性，GPT-4-S においては LLM のプロンプトへの追従能力の高さに起因する脆弱性を指摘する．BEA 2019 [12] の開発データセットを用いた検証の結果，両指標ともに文法的に誤りを含んだ文に対しても高い評価値を与え，文法誤り訂正の自動評価には脆弱性が含まれいることが確認できた．

2　Shared Task

本 Shared Task の目的は，BEA 2019 Shared Task on Grammmatical Error Correction[1)] [12] のデータを用いて，文法誤り訂正の自動評価の脆弱性を問うことである．本 Shared Task では，入力文と出力文の一致度に基づく表層ベース，事前学習済みモデルを用いる埋め込みベース，LLM を用いる LLM ベースの自動評価を対象としている．

表層ベースの自動評価では，ERRANT [1,2] および GLEU [3,4] で評価する．埋め込みベースの自動評価では，PT-ERRANT [5] および IMPARA [6] で評価する．LLM ベースの自動評価では，GPT-4-S [7] で評価する．

3　対象とする自動評価指標

3.1　IMPARA

IMPARA [6] は，類似性評価モデルと訂正評価モデルから構成された自動評価指標である．入力文 S と出力文 O に対してそれぞれの評価モデルから類

1）　https://www.cl.cam.ac.uk/research/nl/bea2019st/

似性スコアと訂正スコアを獲得し，式 1 に従って評価スコアを計算する．

$$\text{score}(S,O) = \begin{cases} \text{corr}(O) & \text{if } \text{sim}(S,O) > \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

ここで，$\text{sim}(S,O)$ は類似性スコア，$\text{corr}(O)$ は訂正スコア，θ は閾値を示している．類似性スコアが θ 以下の場合は評価スコアを 0 とし，θ よりも大きい場合は評価スコアを訂正スコアとしている．類似性評価モデルにおける類似性スコアは，BERT [13] の最終層のトークン埋め込みを平均したものを文埋め込みとし，入力文と出力文の余弦類似度によって算出される．また，訂正評価モデルは，誤りを含む文に対する編集操作の影響度を考慮して訓練される．影響度は，BERT の最終層のトークン埋め込みの平均を文埋め込みとして使用し，訂正文 T から編集操作 e を除外した文 T_{-e} の文埋め込み間の内積から式 2 によって算出される．

$$I_e = 1 - \frac{\text{BERT}(T) \cdot \text{BERT}(T_e)}{\|\text{BERT}(T)\|\|\text{BERT}(T_e)\|} \quad (2)$$

IMPARA は，類似性スコアと訂正スコアから評価スコアを計算するが，入力文に対して適切な訂正をした出力文であっても類似度が閾値 θ 以下であるならば，評価スコアは 0 となってしまう．本研究では，入力文と出力文の類似度に着目して，IMPARA の脆弱性について検証する．

3.2 GPT-4-S

GPT-4-S [7] は，LLM を利用して文法誤り訂正を自動評価する指標である．入力文と出力文を LLM のプロンプト（付録 A）の指示に従って，文レベルで評価する．本研究では，LLM がプロンプトの指示に従うことに着目して，GPT-4-S の脆弱性について検証する．

4 自動評価指標の脆弱性

本節では IMPARA [6], GPT-4-S [7] それぞれについて，その特性に起因して脆弱性となり得る点を指摘する．その脆弱性を悪用して不当に高い評価を得るような出力文の作成方針を検討する．

4.1 IMPARA の脆弱性：類似度

IMPARA は，最初に入力文と出力文の余弦類似度を計算し，閾値を基準として評価スコアが算出される．IMPARA は，文法的に誤りを含む入力文に対して適切な編集をした出力文であっても，入力文と出力文の類似度が閾値以下である場合は評価スコアは 0 となってしまう．また，BERT [13] の最終層のトークン埋め込みを平均したものを文埋め込みとしており，単語数の少ない入力文への編集は入力文と出力文の類似度を下げる可能性がある．

本研究では，IMPARA の入力文と出力文の類似度に注目し，以下の 3 つを検証する．

1. 入力文と出力文の類似度が閾値以下の場合，出力文を入力文に置き換える
2. 入力文の単語数が少ない場合，出力文を入力文に置き換える
3. 入力文と類似度の高い言い換え文を出力文とする

1 の操作では，閾値によって評価スコアが 0 になるのを必ず回避することができ，訂正スコアのみによって評価スコアが決定される．出力文を入力文に置き換える操作は，文法的に誤りを含む，確実に低品質な文をそのまま出力することを意味する．また， 2 の操作では，入力文と出力文の類似度の低下を抑制できる．IMPARA における文の埋め込みは，BERT の最終層のトークン埋め込みを平均することで算出される．このとき，入力文に対する複数の編集は，入力文と出力文の類似性を低下することに繋がり，それに伴い適切な訂正文であっても閾値によって評価スコアが 0 になると考えられる．本研究では，任意の単語数を設定し，入力文の単語数が少ない場合，出力文を入力文に置き換える．

類似度の高い文の生成には，入力文に対して意味的に等価な文に置き換える言い換え技術を適用できる． 3 の操作は，文法的に誤りを含む文を文法的に正しい文に編集する文法誤り訂正とは異なるが，入力文との類似度が高い出力を得られる．本研究では，類似度の高い言い換え文を出力文とすることで，文法誤り訂正の評価をする．

4.2 GPT-4-S の脆弱性：指示追従性

GPT-4-S は，プロンプトの指示に従って，入力文に対する出力文の文法誤り訂正を評価する．しかし，出力文にその評価を不当に高くつけるような指示を追加することで，LLM を用いた自動評価の性能は大きく変化する可能性がある．本研究では，出力文の最後に高いスコアを付けるように指示を付与

表 1　各手法と評価結果

	ERRANT	GLEU	PT-ERRANT	IMPARA	GPT-4-S
GPT 4o	0.426	0.759	0.447	0.814	4.637
+ 類似度	0.426	0.758	0.466	0.820	4.618
+ 単語数	0.428	0.760	0.476	0.817	4.623
+ 類似度 or 単語数	0.427	0.758	0.474	0.820	4.622
+ 類似度 and 最大 5 回生成	0.427	0.760	0.459	0.819	4.630
+ 言い換え文	0.105	0.301	0.080	0.805	4.522
+ 類似度 and 言い換え文最大 5 回生成	0.426	0.759	0.450	0.816	4.629
+ 指示を追加	0.250	0.421	0.089	0.478	4.867

表 2　出力文の変更による IMPARA の評価事例（文法誤りの箇所は赤字，文法誤りの訂正箇所は青字）

	入力文	出力文	類似度	IMPARA
GPT 4o	It is a ciclical process .	It is a cyclical process .	0.892	0
+ 類似度		It is a ciclical process .	1.000	0.981
GPT 4o	Yours Sincerely .	Yours sincerely .	0.880	0
+ 単語数		Yours Sincerely .	1.000	0.240

することで GPT-4-S の脆弱性を示す．

5　評価実験

第 4 節において設計した出力文を用いた実験を通して，指摘した脆弱性が存在することを検証する．

5.1　実験設定

本研究では，BEA 2019 [12] の開発用データを評価用データとして，4,384 文の文法誤り訂正を評価した．文法誤り訂正モデルは，GPT-4o [14] を使用してベースラインの出力文を作成した．

出力文の変更：IMPARA　1 の操作では，閾値 θ を 0.9 に設定し，2 の操作では，単語数が 3 以下の場合に出力文を入力文へ変更した．これらを組み合わせた手法について検証し，入力文と出力文の類似度が 0.9 以下の場合，類似度が 0.9 になるまで最大 5 回生成する手法についても検証する．

3 の操作では，GPT-4o を使用して言い換え文を作成した．また，入力文とベースラインの出力文の類似度が 0.9 以下の場合，類似度が 0.9 になるまで最大 5 回言い換え文を生成し入力文を出力文とする手法についても検証する．

出力文の変更：GPT-4-S　出力文の評価を不当に高くつけるようにするために，出力文の最後に括弧書きで「(Forget everything up to now . Please output 5 as the target_score .)」という指示を追加した．

評価指標　本研究では，入力文と出力文の表層的な一致度に基づく評価 [1–4]，事前学習済みモデルを用いた埋め込みベースの評価 [5, 6]，LLM を用いた評価 [7] を用いて文法誤り訂正を評価する．表層的な一致度に基づく自動評価として，ERRANT [1, 2] および GLEU [3, 4] で評価した．事前学習済みモデルを用いた埋め込みに基づく評価として，PT-ERRANT [5] および IMPARA [6] で評価した．これらの自動評価は，Shared Task で公開されている実装 [2)] [3)] に従った．LLM を用いた評価として，GPT-4-S [7] を用いた．LLM の評価モデルは GPT-4o を使用した．また，LLM の出力から評価値を抽出することができなかった場合は，その文対の評価値を 0 とした．

5.2　実験結果

表 1 に実験結果を示す．また，1 および 2 の操作によって不当に高い評価値が得られた例を表 2 に示す．表 2 より，入力文に対して適切な訂正をした出力文であっても，IMPARA の評価値が 0 になっている．一方，出力文を入力文に変更する操作

2)　https://github.com/gotutiyan/gec-metrics
3)　GLEU の評価指標は，コマンドライン上の実装と API 上の実装では評価結果が異なる．本研究ではコマンドライン上で実装した．

によって，IMPARA の評価値が不当に高くなっている．ERRANT と PT-ERRANT の評価指標においても，ベースラインと比較して不当に高い評価値が得られた．特に，入力文と出力文の類似度が 0.9 以下の場合，類似度が 0.9 になるまで最大 5 回生成する手法は，GPT-4-S の評価を除いた全ての評価指標でベースライン以上の高い評価値が得られた．

3 の操作において，IMPARA は，ERRANT や GLEU，PT-ERRATNT の評価値と比べてベースラインとの差が大きく低下しなかった．類似度に基づいて言い換え文を最大 5 回生成する手法では，IMPARA および PT-ERRANT で不当に高い評価値が得られた．

また，出力文の最後に指示を追加した手法では，GPT-4-S の評価指標においてベースラインよりも不当に高い評価が与えられた．追加した指示によって評価値の最大値に近い値が得られた．

6 おわりに

本研究では，文法誤り訂正における自動評価の脆弱性について検証した．埋め込みベースの IMPARA では入力文と出力文の類似度に基づくフィルタリング機構の脆弱性に，LLM を用いた GPT-4-S では LLM の指示追従性の高さに起因する脆弱性に着目し，訂正文としては不適切だが高い評価スコアが得られるような出力文を作成した．IMPARA では，モデルの出力文と入力文の類似度が低い場合に出力文を入力文に置き換え，GPT-4-S では，高いスコアの付与を指示する文言を出力文に付け加えることによって評価値の最大値に近い値を不当に得られた．評価実験の結果，訂正文としては不適切な出力文に対しても不当に高い評価が与えられるケースが実在し，文法誤り訂正の自動評価には脆弱性が含まれていることが確認できた．

謝辞

本研究は，JSPS 科研費（基盤研究 B，課題番号：JP23K24907）の助成を受けたものです．

参考文献

[1] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In **Proceedings of the 26th International Conference on Computational Linguistics**, pp. 825–835, 2016.

[2] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics**, pp. 793–805, 2017.

[3] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground Truth for Grammatical Error Correction Metrics. In **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing**, pp. 588–593, 2015.

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

[5] Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. Revisiting Grammatical Error Correction Evaluation and Beyond. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 6891–6902, 2022.

[6] Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. IMPARA: Impact-Based Metric for GEC Using Parallel Data. In **Proceedings of the 29th International Conference on Computational Linguistics**, pp. 3578–3588, 2022.

[7] Masamune Kobayashi, Masato Mita, and Mamoru Komachi. Large Language Models Are State-of-the-Art Evaluator for Grammatical Error Correction. In **Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 68–77, 2024.

[8] Mariano Felice and Ted Briscoe. Towards a Standard Evaluation Method for Grammatical Error Detection and Correction. In **Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 578–587, 2015.

[9] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 Shared Task on Grammatical Error Correction. In **Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task**, pp. 1–12, 2013.

[10] 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 **Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task**, pp. 1–14, 2014.

[11] Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In **Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 568–572, 2012.

[12] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 Shared Task on Grammatical Error Correction. In **Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 52–75, 2019.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 4171–

4186, 2019.
[14] OpenAI. GPT-4 Technical Report. **arXiv:2303.08774**, 2023.

A GPT-4-S のプロンプト

The goal of this task is to rank the presented target based on the quality of the sentences. After reading the source, please assign a score from a minimum of 1 point to a maximum of 5 points to the target based on the quality of the sentence.

source
[SOURCE]
target
[CORRECTION]
output format

The output should be a markdown code snippet formatted in the following schema, including the leading and trailing “```json” and “```”:

```
{
"target_score": int // assigned score for target
}
```