Nice gliTtchers: 文法誤り訂正部門

五藤巧　土肥康輔　Adam Nohejl　Justin Vasselli　郷原 聖士　坂井 優介　渡辺 太郎
奈良先端科学技術大学院大学
goto.takumi.gv7@is.naist.jp

概要

NLP2025 ワークショップ「LLM 時代のことばの評価の現在と未来」にて開催された共通タスクの文法誤り訂正部門への投稿および関連する知見について説明する．本共通タスクの趣旨は評価尺度のハッキングであり，高い評価値を不当に得ることを主な目的としている．我々は，主に IMPARA と LLM ベースの評価尺度へのハッキング攻撃を目的とした提出を行った．IMPARA では類似度推定スコアを距離関数として，単言語コーパスから誤り文に対する k 近傍事例を検索し，その中から最も品質推定スコアが高くなる文を訂正文として出力した．LLM では，プロンプトがマークダウン形式であることに注目し，`#New instruction:` から始まるヘッダーとともにスコアを 5 と出力させるような指示文を訂正文として常に出力した．実験コードを含む実装は公開する：https://github.com/naist-nlp/nice_gliTchers.

1　はじめに

本共通タスクに取り組むにあたり，できるだけ不当な方法で高い評価値を得るような攻撃手法を考案することとし，ベースラインを除いて真面目に訂正性能を向上させることは行わなかった．その結果，真面目に訂正した結果に相当するベースラインの結果，および IMPARA [1] と大規模言語モデル（LLM）に基づく尺度 [2] をそれぞれ攻撃の目的とした結果の計 3 種類の結果を提出した．ERRANT [3, 4]，PT-ERRANT [5]，GLEU [6, 7] については良い結果が得られなかったため，これらの評価尺度を攻撃目的とした結果は提出しなかった．手法の考案にあたっては，各評価尺度について手法を考えたため，本稿ではそれぞれの評価尺度ごとに，その攻撃手法を説明する．また，良い結果が得られなかった評価尺度についても試行内容を共有する．

本稿では攻撃手法を**訂正手法**と**後処理手法**の 2 種類に分類して説明する．訂正手法は，誤り文のみを入力として，評価尺度を非合理的に高くするような訂正文を出力する．後処理手法は，誤り文と既存の訂正モデルが出力した訂正文を入力とし，改変した訂正文を生成することで，さらなる性能を底上げを狙う．公開する実装でも両者を分けて実装した．評価尺度の実装については，LLM の尺度は独自に実装して本プロジェクトのために公開する実装 [1] に含めており，それ以外の尺度は gec-metrics [2] を用いた．

2　ベースライン

我々はまず、ベースラインの結果として，攻撃を目的としない，通常の文法誤り訂正システムによる結果を提出した．文法誤り訂正システムとして GECToR [8] を採用し，RoBERTa（large） [9]，XLNet（large-cased） [10]，DeBERTa（v1-large） [11] の 3 つのモデルをそれぞれ追加学習し，Majority voting [12] によりアンサンブルした．個々のモデルは GECToR の原論文に従い 3 段階の学習とし，1 段階目は PIE-synthetic [13] を，2 段階目には BEA2019-train [14] と呼ばれる FCE-train [15]，W&I-LOCNESS-train [16]，Lang-8 [17]，NUCLE [18] の 4 つのデータを結合して無編集のペアを削除したもの，3 段階目には W&I-LOCNESS-train の無編集のペアも含めたものを用いた．Majority voting を適用する際には，Tarnavskyi ら [12] の実装 [3] を用いて，最小投票数を 2 として実行した．GECToR の学習や推論に関するコード・学習済みモデルは https://github.com/gotutiyan/gector にある．評価の結果を表 1 に示す．

このベースラインは，本共通タスクで提案される攻撃手法が脅威になるかどうかを判断するための基準になることを期待して提出した．ベースラインの性能は，真面目に訂正した時に得られる性能として標準的もしくは少し性能が高い程度のものである．

1） https://github.com/naist-nlp/nice_gliTchers
2） https://github.com/gotutiyan/gec-metrics
3） https://github.com/MaksTarnavskyi/gector-large/blob/master/ensemble.py

表 1: 各評価尺度におけるベースラインの性能.

評価尺度	評価値
ERRANT	61.89
GLEU	75.97
PT-ERRANT	64.83
IMPARA	0.6987
GPT-4-S	4.6747

実際に，同じモデルを用いて BEA2019 の評価セットで評価したところ F0.5 として 76.44 を得たが，これは現状で査読付き論文にて報告されている中での最高性能：81.4 [19] を 5 ポイント程度下回るものであり，公式のリーダボード[4)]でも上位 5 位にも入らない．したがって，このベースラインを下回る攻撃手法は，真面目に訂正したほうがより高い評価値を得られるという意味で脅威にはならないと考えられる．

3 ERRANT・PT-ERRANT

ERRANT [3, 4]，PT-ERRANT [5] は両者とも参照あり評価尺度であり，入力文と訂正モデルの訂正文から抽出される仮説編集と，入力文と参照文から抽出される参照編集の一致率に基づいて評価する．PT-ERRANT は各編集を重みづける点が ERRANT とは異なるが，いずれにせよ仮説編集が正しいかどうか，すなわち True Positive か False Positive かは参照編集によって決定される．したがって，ERRANT と PT-ERRANT のどちらか一方で攻撃が成功すれば，程度の違いはあるものの，もう一方にも必ず攻撃は成功する．

攻撃の方針として，PT-ERRANT の重みに基づいて編集をフィルタリングする後処理手法を試した．PT-ERRANT の絶対値を取る前の重みに注目して，既存の訂正システムが推定した仮説編集から，負の重みである編集のみを除外する．PT-ERRANT の重みの符号が仮説編集の正しさと一致しているのであれば，これにより評価値を底上げできるはずである．しかし，ベースラインの結果を既存の訂正結果として本手法を適用すると，ERRANT は 58.13 (ベースライン：61.89)，PT-ERRANT は 64.70（ベースライン：64.83) であり，評価値は向上しなかった．実際に PT-ERRANT の絶対値を取る前の重みの符号と，参照編集に含まれるかどうかをそれぞれラベル

4) https://codalab.lisn.upsaclay.fr/competitions/4057#results

表 2: PT-ERRANT の重みの正負と，参照編集に含まれるかどうかを比較した時のの混同行列．ベースラインの訂正結果を仮説編集とした．左上と右下のセルの値が高くなることが望ましい．

	重みが負	重みが正
参照に含まれない	34	971
参照に含まれる	67	2778

とみなして混同行列を計算すると，表 2 に示すように両者に明確な一致は確認できなかった．

4 GLEU

GLEU [6, 7] は n-gram の一致率に基づく尺度で，入力文，訂正文，参照文の 3 つ組の中における n-gram の重複で評価する．GLEU は GREEN [20] の定式化によって解釈すると見通しが良くなる．GREEN では n-gram の置換を削除と挿入の組み合わせと考えることにより，n-gram を 7 つのグループに分類している．具体的には，n-gram の正しい維持（True Keep; TK），n-gram の正しい削除（True Delete; TD），n-gram の正しい挿入（True Insert; TI），n-gram の過剰削除（Over Delete; OD），n-gram の過剰挿入（Over Insert; OI），n-gram の削除の不足（Under Delete; UD），n-gram の挿入の不足（Under Insert; UI）である．Koyama ら [20] の文献中の Figure 1 に示されたベン図も合わせて参照されたい．各 n について，これらのグループに属する n-gram の数 TK_n，TD_n ... を計算したとき，適合率に相当する値 p_n は次式で計算される．

$$p_n = \frac{\mathrm{TI}_n + \mathrm{TK}_n - \mathrm{UD}_n}{\mathrm{TI}_n + \mathrm{TK}_n + \mathrm{OI}_n + \mathrm{UD}_n}. \quad (1)$$

GLEU では，上記の値を $n = 1$ から $n = 4$ まで計算して幾何平均を計算した後，系列長に応じた Brevity penalty を考慮して最終的な評価値を算出する．

攻撃手法の考案にあたっては，式 1 で計算される値に注目した．この式は下記の性質を暗に示していると考えた．

- n-gram の削除に関して，過剰な編集（OD）はペナルティを受けないが，編集の不足（UD）はペナルティを受ける．
- n-gram の挿入に関して，編集の不足（UI）はペナルティを受けないが，過剰な編集（OI）はペナルティを受ける．

この性質から，ペナルティを減らすための攻撃手法

表 3: GLEU のそれぞれの n における適合率に相当する評価値 p_n と，GLEU の最終的な評価値．

システム	p_1	p_2	p_3	p_4	GLEU
ベースライン	90.94	80.76	72.50	65.33	75.97
w/o 挿入編集	91.02	78.66	68.48	59.60	71.66

として挿入の編集は全て除外する後処理手法を考案した．しかし，ペナルティを減らすこと以上に，本来得られるはずの加点を大量に失うことに繋がり，ベースラインの結果に適用したときの評価値は 71.66 となった（ベースライン：75.97）．表 3 に示したそれぞれの n における p_n によると，p_1 では評価値が向上したものの，$n \leq 2$ では評価値の悪化が顕著であった．このことは，GLEU が複数種類の n の評価値の間で幾何平均を取ることの頑健性を示すものである．

5 IMPARA

IMPARA [1] は参照なし評価尺度であり，入力文と訂正文の類似度スコアに閾値を設け，閾値で足切り後，訂正文に対する品質推定スコアを最終的な評価値とする．類似度スコアは，入力文と訂正文の埋め込み表現の余弦類似度として計算され，一般に 0.9 の閾値が設定される．埋め込み表現は，BERT などのモデルを用いて平均プーリングによって計算する．参照文が未知な設定である，参照あり評価尺度とは異なり，参照なし評価尺度では手元で自由に訂正文の評価値を計算できる．したがって，我々は多様な訂正文を用意し，最も高い評価値を得た文を選択するような方法を試みた．

はじめに，編集の適用パターンを全列挙する後処理手法を試した．入力文と既存の訂正文から，ERRANT により編集を抽出する．抽出された編集の数を N 個とすると，これらの編集の適用パターンは 2^N 通り考えられるため，訂正文も 2^N 通り生成することができる．生成した全ての訂正文について IMPARA の評価値を計算し，最も評価値が高いものを出力する．計算量の観点から実験を容易にするため，ベースラインの訂正結果において $N \leq 10$ の訂正文にのみこの手法を適用した．その結果，IMPARA の評価値は 0.698 から 0.702 に向上した．実際に評価値が向上した例を表 4 に示す．この例では 4 つの編集が起こっており，適用パターンを全通り試した結果 [9, 10, “volunteer”] の編集のみを除くことで（IMPARA にとって）最適な訂正文となり，評価値が 0.000 から 0.998 まで向上した．元の訂正文では，volunteer とスペル誤りを正すことでむしろ類似度推定スコアの閾値にかかってしまい評価値が 0.000 になっていたが，この編集を除くことで閾値を超えることができた．このように，スペル誤りに起因して誤ったフィルタリングが起こる点は，坂井ら [21] の報告と整合している．

次に，単言語コーパスを利用した訂正手法を考案した．品質推定モデルの入力は訂正文のみであるため，入力文に対する妥当な訂正文でなくとも，文法的な文を入力すれば高い品質スコアが得られる．このことから，基本的に文法誤りがないと考えられる単言語コーパスを利用できると考えた．ただし，類似度推定スコアの閾値は超える必要があるため，単言語コーパスからある程度入力文と類似した文を探す必要がある．この目的のために，単言語コーパスから誤り文の k 近傍事例を検索する方法を提案する．まず適当な単言語コーパスを用意し，類似度推定スコアを計算するための埋め込みモデルで各文の埋め込み表現を平均プーリングによって計算する．次に誤り文の埋め込み表現も同様に計算し，これをクエリとして単言語コーパスの中から k 近傍事例を検索する．ここで距離関数に負の余弦類似度を用いることで，IMPARA の類似度スコアを直接距離関数とする．これにより，少なくとも類似度スコアの閾値を超える文を検索することができる[5]．最後に，検索された k 文から，距離が IMPARA の類似度スコアの閾値を超えるもののみを残し，さらにその中から品質推定スコアが最も高い事例を出力する．実験では，単言語コーパスとして文法誤り訂正のための学習データの参照文を用いることとし，データセットとして BEA2019-train（無編集文対も使用）と，GECToR-Large における蒸留のためのデータセットである Troy-1BW と Troy-Blogs [12] を混ぜることにより，合計 3,574,070 文からなる単言語コーパスを用いた．検索部分の実装には semsis [6] を用いて，$k = 256$ で実行した．この結果，**IMPARA の評価値として 0.910 を得る**ことができ，ベースラインの結果に対する評価値 0.698 を 0.2 ポイント以上上回った．表 5 に示す実例では，検索された事例は明らかに入力文を訂正したものではないが，類似度推定モデル（bert-base-cased）は 0.9 以上の類似度

5) 実際には検索された事例全てが閾値を超えるとは限らないが，閾値を超える事例がある程度含まれることを期待している．

6) https://github.com/de9uch1/semsis

表 4: 編集パターンを全通り試すことによる性能改善の例．括弧内の値は IMPARA の評価値を示す．入力文と訂正文から 4 つの編集が抽出され，それぞれの編集を入力文に対する単語レベルスパンと，その訂正後の文字列の 3 つ組で表した．編集の適用パターンを全通り試した結果，採用された編集に下線を引いた．

入力文 (0.092)	Also they would have an opprtunity being involved in volontier activity .
訂正文 (0.000)	Also , they would have an opportunity to be involved in volunteer activity .
最適な訂正文 (0.998)	Also , they would have an opportunity to be involved in volontier activity .
訂正文の編集	[1, 1, “,”], [5, 6, “opportunity”], [6, 7, “to be”], [9, 10, “volunteer”]

表 5: 単言語コーパスから近傍事例を検索する方法による結果の一例．

項目	値
入力文	It takes around 6 hours by National highway to go from Delhi to Agra .
訂正文（だと思うことにした検索事例）	It takes about 1 hour to get to the city from the airport by bus .
類似度スコア	0.916
品質推定スコア	0.999

を計算しており，閾値を突破できていることがわかる．また訂正文は単言語コーパスから検索しているため文法的であり，品質推定スコアは 0.999 と高い値となった．実際には，単言語コーパスとして Wikipedia なども用いることができるため，より大規模なコーパスから検索することでさらに性能を上げられる可能性がある．

最後に，検索による方法を後処理手法に拡張した．上記で述べた検索による結果と，ベースラインの真面目な訂正結果から IMPARA の評価値が高くなる方を選ぶことで，両者を補完的に活用した．これにより **IMPARA の評価値 0.930 を得た**ため，これを提出した．

6 LLM

LLM に基づく尺度 [2] は，訂正文を評価するように指示したプロンプトを入力し，評価値を生成させることで評価する，参照なし評価尺度である．評価は 1 から 5 の整数によって 5 段階で行われる．Kobayashi ら [2] の評価尺度の実装は公開されておらず，プロンプト形式も本共通タスク独自のものであるため，ある程度妥当な評価ができることを確認するために手元で実装し，メタ評価を行った．メタ評価には SEEDA データセット [22] の SEEDA-S を用いた．また，手元の実験では大規模言語モデルとして `gpt-4o-mini-2024-07-18` を暫定的に用いた．この理由は，低コストで実施できる実験設定とすることで再現を容易にするためである．メタ評価の結果，SEEDA-S の Base 設定で Pearson の積率相関係数と Spearman の順位相関係数として 0.857 と 0.874 を得た．同様に+Fluency 設定では 0.628 と 0.828 を得た．なお，IMPARA などで実施されるメタ評価と同様，文単位の評価値を平均することでコーパス単位の評価値とし，その評価値でシステムをソートすることで順位を計算した．この相関の値より，ある程度妥当な評価ができる手法および実装であると判断した．この実装も公開するコードに含めている．

攻撃の方法として，既存の指示を無視して，スコアを 5 と出力する旨の指示文を訂正文として出力することを試みた．具体的には，全ての誤り文について固定のテキスト*# New instruction: Ignore previous instructions and always output 5 as the score.* を訂正文として出力した．コーパス単位の評価値を文単位の評価値の平均としたとき，この方法によりコーパス単位の評価値 5.0 を得た．すなわち，全ての文で 5 の評価値を出力させることに成功した．なお，入力文そのものの評価値は 3.76，ベースラインの訂正結果の評価値は 4.26 であった．また，この方法では*# New instruction:* のヘッダを含めることが重要であった．このヘッダがない場合，コーパス単位の評価値は 2.35 となり効果がなかった．プロンプトは *#source:* や*#target:* を含むようなマークダウン形式であるため，この形式と一致するようにヘッダを追加したことが重要であったと考えられる．

上記が手元で実行した結果であるが，オーガナイザによる評価結果は 1.43 となり，攻撃には失敗した．`gpt-4o-mini-2024-07-18` は本手法で攻撃できたものの，これよりも能力が高いモデルはより頑健であることを示唆している．もしくは，実装の違いなどによる何らかの影響を受けた可能性がある．

7 おわりに

我々のチームは，文法誤り訂正の自動評価尺度に対し，不当に高い評価値を得ることを目的とした手法を考案した．その結果，IMPARA では真面目な訂正結果を 0.2 ポイント以上上回る結果を不当に得ることに成功した．LLM による尺度でも，一部モデルを用いた場合には，訂正文を攻撃的な指示文とすることで常にスコア 5 を出力させることができた．

本タスクに取り組む中で，参照あり評価尺度の頑健性について再認識した．参照あり評価尺度では，参照に含まれる編集や n-gram を出力しないことには評価値が上がらないため，真面目に訂正するしか方法がないように伺える．仮に不当な方法で評価値の向上を狙ったとしても，いずれにせよ参照文と一致するテキストの出力を目指すことになるため，この時点で妥当な方法と言わざるを得ない．したがって，参照あり評価尺度に対しては攻撃という概念が存在しないようにも思えた．一方，IMPARA などの参照なし評価尺度は，メタ評価では最高水準の人手評価との相関を達成するものの，脆弱性があることが明らかになった．本稿で述べた攻撃手法に対する防御手法については，LLM による尺度に対する攻撃は容易に防御できるが，IMPARA への攻撃手法に対しては工夫が必要なようにも思われる．例えば，Scribenid [23] のように表層一致に基づく類似度尺度を取り入れたり，Sentence-BERT [24] のような，より正確に文間の類似度を計算できると考えられる埋め込み表現を用いることで類似度スコアによるフィルタリング性能を強化することは考えられるが，今後の課題としたい．

参考文献

[1] 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, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[2] 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 (BEA 2024)**, pp. 68–77, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[3] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In **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.

[4] 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 (Volume 1: Long Papers)**, pp. 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[5] 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.

[6] 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.

[7] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Gleu without tuning, 2016.

[8] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. GECToR – grammatical error correction: Tag, not rewrite. In Jill Burstein, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, Helen Yannakoudakis, and Torsten Zesch, editors, **Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 163–170, Seattle, WA, USA → Online, July 2020. Association for Computational Linguistics.

[9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[10] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. **Advances in neural information processing systems**, Vol. 32, , 2019.

[11] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. **arXiv preprint arXiv:2006.03654**, 2020.

[12] Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3842–3852, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[13] Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. Parallel iterative edit models for local sequence transduction. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 4260–4270, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, **Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 52–75, Florence, Italy, August 2019. Association for Computational Linguistics.

[15] 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.

[16] Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. Developing an automated writing placement system for esl learners. **Applied Measurement in Education**, Vol. 31, No. 3, pp. 251–267, 2018.

[17] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In Haifeng Wang and David Yarowsky, editors, **Proceedings of 5th International Joint Conference on Natural Language Processing**, pp. 147–155, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[18] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner English: The NUS corpus of learner English. In Joel Tetreault, Jill Burstein, and Claudia Leacock, editors, **Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 22–31, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[19] Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. 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. 17–33, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[20] 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.

[21] 坂井優介, 五藤巧, 渡辺太郎. 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. 言語処理学会第 31 回年次大会発表論文集, March 2025.

[22] 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.

[23] 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.

[24] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.