Grammatical Error Correction Evaluation by Optimally Transporting Edit Representation

Takumi Goto Yusuke Sakai Taro Watanabe
Nara Institute of Science and Technology, Japan
{goto.takumi.gv7, sakai.yusuke.sr9, taro}@is.naist.jp

Abstract

Automatic evaluation in grammatical error correction (GEC) is crucial for selecting the best-performing systems. Currently, reference-based metrics are a popular choice, which basically measure the similarity between hypothesis and reference sentences. However, similarity measures based on embeddings, such as BERTScore, are often ineffective, since many words in the source sentences remain unchanged in both the hypothesis and the reference. This study focuses on edits specifically designed for GEC, i.e., ERRANT, and computes similarity measured over the edits from the source sentence. To this end, we propose *edit vector*, a representation for an edit, and introduce a new metric, UOT-ERRANT, which transports these edit vectors from hypothesis to reference using unbalanced optimal transport. Experiments with SEEDA meta-evaluation show that UOT-ERRANT improves evaluation performance, particularly in the +Fluency domain where many edits occur. Moreover, our method is highly interpretable because the transport plan can be interpreted as a soft edit alignment, making UOT-ERRANT a useful metric for both system ranking and analyzing GEC systems. Our code is available from https://github.com/gotutiyan/uot-errant.

1 Introduction

Grammatical Error Correction (GEC) task aims to automatically correct grammatical errors found in an input sentence regarding various grammatical items, such as verb tenses, noun numbers, and spelling mistakes. Many GEC systems have been proposed (Bryant et al., 2019; Omelianchuk et al., 2020; Rothe et al., 2021; Tarnavskyi et al., 2022; Katinskaia and Yangarber, 2024; Omelianchuk et al., 2024) with diversity in performance, and users need to select the system suitable for their purposes by referring to the results of automatic evaluation metrics. Considering that large language models are launched frequently and can also be used as correction systems, the role of automatic evaluation metrics is increasingly important for system selection.

A typical GEC metric is reference-based evaluation, which compares a hypothesis sentence output from a system with a reference sentence annotated by humans. The central idea of the reference-based evaluation is to quantify the similarity between the hypothesis and the reference. Considering the recent development of neural models, evaluation could be carried out based on the similarity of contextualized word representation, such as BERTScore (Zhang et al., 2020). However, since only tokens that are identified as errors are edited in GEC, many words remain unchanged in the hypothesis and reference. For this reason, the information of matching tokens becomes dominant in the sentence or word embeddings based metrics, which does not lead to a good evaluation metric (Gong et al., 2022).

In this study, we solve this problem by introducing the similarity of edits. An edit represents the difference between an input sentence and its corrected version as a rewrite of fine-grained units, such as tokens or phrases, and can be automatically extracted using tools like ERRANT (Felice et al., 2016; Bryant et al., 2017). By using edits as proxies, it is possible to compare the hypothesis and reference sentences not directly, but as changes from the input sentence. Inspired by methods that quantify the similarity of sentences or words as the similarity of their embeddings, this study proposes to measure similarity based on the embedding representation of edits.

For this purpose, first, we propose an *edit vector*, which is a vectorized representation of an edit measuring the impact on the sentence representa-

Transactions of the Association for Computational Linguistics, vol. 14, pp. 1723–1741, 2026. https://doi.org/10.1162/TACL.a.747
Action Editor: Tanmoy Chakraborty. Submission batch: 8/2025; Revision batch: 1/2026; Published 7/2026.

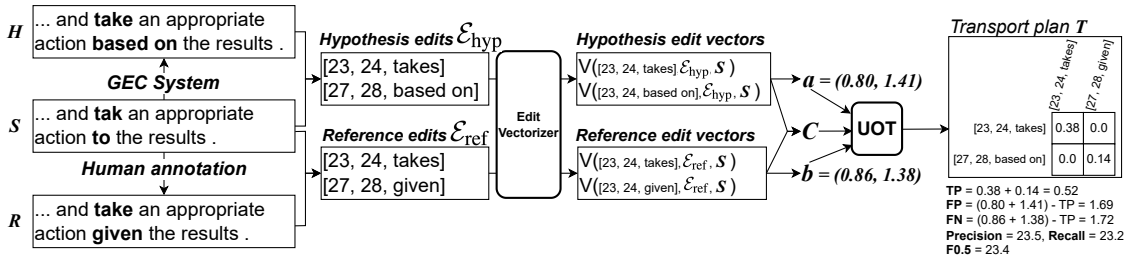


Figure 1: An overview of the proposed metric, UOT-ERRANT. Edits are extracted from the hypothesis and the reference, respectively, and converted into edit vectors. The optimal transport plan of edit vectors from the hypothesis to the reference vectors is then decomposed into a precision, recall, and $F_{0.5}$ score.

tion, defined as the difference vector between the embedding of the complete corrected sentence after applying all edits from an input sentence and a corrected sentence but reverting the corresponding single edit. Next, we apply this to GEC evaluation by comparing the hypothesis and reference edit vectors using unbalanced optimal transport (UOT) (Frogner et al., 2015). In UOT, we transport the norm of edit vectors from the hypothesis to reference using the cost matrix defined over the L^2 distances of edits of two sentences. L^2 distance of two edits of a hypothesis and reference pair measures the closeness of edits, while the norm of an edit vector measures an impact on the corrected sentence of the single edit. The more transported values from the hypothesis to the reference imply more credit to the hypothesis. Finally, by decomposing the transportation plan into precision, recall, and F_β, we adapt the UOT output to the general evaluation method of GEC. The overview shown in Figure 1 explains the process of the proposed metric: extraction of edits, vectorization, transportation by UOT, and decomposition of the transportation plan into evaluation values. Our proposed metric, UOT-ERRANT, relies on edits from ERRANT, but could be regarded as a soft-variant of edits when compared to M^2 (Dahlmeier and Ng, 2012) and ERRANT without strict surface form matching.

In our experiments, we mainly evaluate the performance of UOT-ERRANT on SEEDA (Kobayashi et al., 2024b), a representative meta-evaluation benchmark, and achieve results that surpassed conventional metrics, such as ERRANT (Felice et al., 2016; Bryant et al., 2017) and PT-ERRANT (Gong et al., 2022), in correlation with human judgments. In particular, we demonstrated that evaluation performance is improved when requiring a smaller number of edits, and further gains are observed when many edits are involved by considering edits in contexts. We analyze the properties of the edit vector, focusing on error types (Bryant et al., 2017). We show that edits related to content words have larger norms and that they form clusters according to error type. These result indicates that the edit vector appropriately captures textual changes.

2 Related Work

Edit-level GEC Evaluation Edit-level metrics are the most representative ones for evaluating GEC systems, such as M^2 (Dahlmeier and Ng, 2012), ERRANT (Felice et al., 2016; Bryant et al., 2017), and PT-ERRANT (Gong et al., 2022). Beyond word-wise edit metrics, chunk-wise units have also been proposed in prior studies (Gotou et al., 2020; Ye et al., 2023). These metrics can be interpreted as defining the similarity between a hypothesis sentence and a reference sentence as an edit or chunk matching rate. Essentially, an edit holds information about a span in the input sentence and its corresponding edited string. Existing metrics calculate the match between hypothesis edits and reference edits based on a complete match of this information. This evaluation method is considered a *hard* alignment-based metric since it relies on a binary decision, i.e., a hypothesis edit is correct if it is included in the reference edits, and incorrect otherwise. Hard alignment judges edits as incorrect solely based on differences in their surface forms, even when their impact for the semantic change is semantically accurate.

Quantifying Edits Several methods have been proposed to quantify the impact of edits for GEC evaluation. PT-ERRANT (Gong et al., 2022) employs BERTScore (Zhang et al., 2020) to measure the different of a single edit, which is used

as an edit weight to improve the evaluation performance of edit-level metrics. IMPARA (Maeda et al., 2022) calculates the cosine similarity of the sentence representations between a corrected sentence and a sentence missing a single edit. It is used to generate pseudo-labels for training the quality estimation model, leading to high evaluation performance. Quantifying edits as a scalar in these studies limits their expression ability and prevents their use in similarity calculations. Indeed, PT-ERRANT relies on surface-level matches for edit alignment, with the scalar's role being solely for edit weighting.

NLP Applications of Optimal Transport Optimal transport (Kantorovich, 1958) has been applied to various NLP evaluation methods by quantifying the similarity between distributions. By treating words as discrete distributions based on their Word2Vec (Mikolov et al., 2013) or BERT-based word representations (Devlin et al., 2019), the similarity between sentences is quantified based on a transport cost between words, known as Word Mover's Distance (Kusner et al., 2015). The quantified similarity has been applied to tasks such as nearest neighbor search (Kusner et al., 2015; Yurochkin et al., 2019) and evaluation by measuring the similarity between hypotheses and references (Kilickaya et al., 2017; Chow et al., 2019; Zhao et al., 2019; Colombo et al., 2021). Furthermore, the calculated transport plan can be interpreted as alignment between samples and this property has been used for various tasks such as word alignment (Arase et al., 2023), text matching (Swanson et al., 2020), and word-level translation quality estimation (Kuroda et al., 2024).

3 Proposed Methods

3.1 Edit Vector

We propose an *edit vector*, a vectorization method for an edit which takes into account the impact of the change toward sentence-level semantic representation. Given an original sentence S and its edited version H, we extract a set of edits $\mathcal{E} = \{e_i\}_{i=1}^{|\mathcal{E}|}$ using an edit extraction tool such as ERRANT (Felice et al., 2016; Bryant et al., 2017). Next, for each individual edit $e \in \mathcal{E}$, the edit vector $V(e, \mathcal{E}, S)$ is calculated as the difference between the sentence representation of the complete edited sentence $S_{\mathcal{E}}$ and the corrupted variant $S_{\mathcal{E}\setminus\{e\}}$ that lacks the edit e as follows:

$$V(e, \mathcal{E}, S) = \mathrm{Enc}(S_{\mathcal{E}}) - \mathrm{Enc}(S_{\mathcal{E}\setminus\{e\}}), \quad (1)$$

where $S_{\mathcal{E}}$ is a sentence after applying editing set $\mathcal{E}$ to S. $\mathrm{Enc}(\cdot) \in \mathbb{R}^d$ is a sentence encoder that embeds a sentence into a d-dimensional representation, such as a mean pooling of BERT (Devlin et al., 2019) representation.

The edit vector represents how much the sentence semantic changes by applying an edit. The direction and length of the vector can be interpreted as quantifying how and to what extent an edit changes the sentence meaning, respectively. Because of this, we expect edits that bring about similar semantic changes to be encoded as similar vectors. Compared to methods that quantify an edit as a scalar (Gong et al., 2022; Maeda et al., 2022), edit vectors offer a richer representation of an edit's impact, enabling operations such as similarity calculations.

3.2 Edit-level Similarity via Optimal Transport

3.2.1 Discrete Optimal Transport

Optimal Transport is the problem of finding a transport plan to move one distribution to another. The inputs are two discrete distributions $\boldsymbol{a} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^m$, and a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n\times m}$ between samples. The cost matrix C_{ij} corresponds to the cost of transporting from a_i to b_j. The output is the optimal transport plan $\mathbf{T} \in \mathbb{R}_+^{n\times m}$, where T_{ij} denotes the amount of transport from sample a_i to sample b_j. This plan minimizes the product of transport amounts and costs, $\sum_{ij} T_{ij}C_{ij}$. The plan $\mathbf{T}$ can also be interpreted as an alignment between samples, where samples with more transport are considered to be more strongly aligned. To prevent the trivial solution where $\mathbf{T}$ becomes a zero matrix, which would minimize $\sum_{ij} T_{ij}C_{ij}$ without meaningful transport, additional constraints are generally added to ensure that meaningful transport occurs.

Balanced Optimal Transport (BOT) BOT (Kantorovich, 1958) adds the constraints that all of $\boldsymbol{a}$ must be transported, and $\boldsymbol{b}$ must receive all mass. Therefore, it is assumed that $\boldsymbol{a}$ and $\boldsymbol{b}$ have the same total mass, i.e., $\sum_{i=1}^{n} a_i = \sum_{i=1}^{m} b_i$. For example, probability distributions satisfy this condition. Formally, these constraints can be formulated as a combinatorial optimization

problem:

$$\mathrm{OT}(\boldsymbol{a}, \boldsymbol{b}, \mathbf{C}) = \underset{\mathbf{P} \in \pi(\boldsymbol{a}, \boldsymbol{b})}{\operatorname{argmin}} \sum_{i,j} P_{ij} C_{ij}, \quad (2)$$

$$\text{where } \pi(\boldsymbol{a}, \boldsymbol{b}) = \{\mathbf{X} \in \mathbb{R}_{+}^{n \times m} : \mathbf{X}\mathbb{1}_m = \boldsymbol{a}, \mathbf{X}^{\intercal}\mathbb{1}_n = \boldsymbol{b}\}. \quad (3)$$

Here, $\mathbb{1}_n$ is an n-dimensional vector with all elements are 1. This combinatorial optimization problem can be solved by an efficient solver (Pele and Werman, 2009).

Unbalanced Optimal Transport (UOT) UOT (Frogner et al., 2015) is a relaxed version of the BOT constraints. While BOT assumes that all samples between distributions are transported, this might not be appropriate for certain tasks. For example, in word alignment, it is not assured that all words may match, but spurious words in a sentence are left unaligned (Arase et al., 2023). However, the transport plan generated by BOT forces the transport of all samples, making it unable to handle such unaligned words. To resolve this issue, UOT replaces the $\mathbf{X}\mathbb{1}_m = \boldsymbol{a}$ and $\mathbf{X}^{\intercal}\mathbb{1}_n = \boldsymbol{b}$ terms in the BOT constraints with regularization terms. Formally, we seek a transport plan that minimizes the following objective function:

$$\mathrm{UOT}(\boldsymbol{a}, \boldsymbol{b}, \mathbf{C}) = \underset{\mathbf{P} \in \mathbb{R}_{+}^{n \times m}}{\operatorname{argmin}} \sum_{i,j} C_{ij} P_{ij} + \epsilon H(\mathbf{P}) + \lambda_1 \mathrm{KL}(\mathbf{P}\mathbb{1}_m, \boldsymbol{a}) + \lambda_2 \mathrm{KL}(\mathbf{P}^{\intercal}\mathbb{1}_n, \boldsymbol{b}). \quad (4)$$

Here, $H(\cdot)$ is an entropy regularization term, and $\mathrm{KL}(\cdot)$ is the Kullback-Leibler divergence. Increasing the parameter ϵ leads to a more uniform transport between samples. Likewise, increasing λ_1 and λ_2 encourages transportation of all mass. This regularized optimization problem can be solved using the Sinkhorn algorithm (Sinkhorn, 1964; Chizat et al., 2017).

3.2.2 Connection between UOT and GEC

We apply the UOT framework to measure edit-level similarity. The fundamental idea is to transport hypothesis edit vectors to reference edit vectors, and UOT is applied by treating $\boldsymbol{a}$ as the hypothesis edits, $\boldsymbol{b}$ as the reference edits, and $\mathbf{C}$ as the cost matrix based on the distances between the hypothesis and reference edit vectors. The more mass that can be transported, the better the hypothesis edits align with the reference edits, and thus the higher the similarity. The reason for using UOT constraints is that GEC systems often result in over-correction or under-correction, leading to null-alignments between edits. For instance, in over-correction, some of hypothesis edits become null-alignment, while in under-correction, some of reference edits become null-alignment. Compared to existing studies that primarily focus on word transport (Zhao et al., 2019; Colombo et al., 2021), this study proposes edit transport.

Formally, given a source sentence S, a hypothesis sentence H, and a reference sentence R, we extract a set of hypothesis edits $\mathcal{E}_{\text{hyp}}$ and a set of reference edits $\mathcal{E}_{\text{ref}}$. Tools such as ERRANT can be used to extract these edits, and we convert each of extracted edits into an edit vector.

$$\mathbf{V}^{\text{hyp}} = \{V(e, \mathcal{E}_{\text{hyp}}, S) | e \in \mathcal{E}_{\text{hyp}}\} \quad (5)$$

$$\mathbf{V}^{\text{ref}} = \{V(e, \mathcal{E}_{\text{ref}}, S) | e \in \mathcal{E}_{\text{ref}}\} \quad (6)$$

We define $\boldsymbol{a}$ and $\boldsymbol{b}$ as the norm of edit vectors $\mathbf{V}^{\text{hyp}}$ and $\mathbf{V}^{\text{ref}}$, respectively, and $\mathbf{C}$ as the Euclidean distance between edit vectors. As discussed in Section 3.1, each edit vector represents the impact toward the change in sentence semantic with norm representing as its magnitude. Thus, using the norm for $\boldsymbol{a}$ and $\boldsymbol{b}$ is regarded as an implicit weighting of edits. We assume that the Euclidean distance is an appropriate measure for the transport cost of edits, based on the intuition that similar edits are encoded into vectors with similar directions and norms. Finally, we compute the optimal transport plan $\mathbf{T} = \mathrm{UOT}(\boldsymbol{a}, \boldsymbol{b}, \mathbf{C})$. This plan represents edit-level alignment, and the total transported mass $\sum_{ij} T_{ij}$ is regarded as a similarity between two edit sets.

3.3 Proposed Metric: UOT-ERRANT

As an application of the edit-level transport plan, we propose a new metric for GEC, namely *UOT-ERRANT*. UOT-ERRANT calculates precision, recall, and F_β, which are typically used in GEC, by focusing on both the transported and non-transported mass from the transport plan. This quantifies the degree of over-correction and under-correction in addition to the simple similarity between the hypothesis and the reference. Figure 1 shows an example where both the hypothesis and the reference have two edits.

First, we decompose the transport plan $\mathbf{T}$ into True Positive (Score_{TP}), False Positive (Score_{FP}),

and False Negative (Score_{FN}). Score_{TP} is the degree to which hypothesis edits match reference edits, calculated as the sum of transported mass:

$$\text{Score}_{\text{TP}} = \sum_{i,j} T_{ij}. \tag{7}$$

Score_{FP} corresponds to the hypothesis edits that are considered incorrect, calculated as the sum of mass that could not be transported from the hypothesis:

$$\text{Score}_{\text{FP}} = \sum_{i} a_i - \text{Score}_{\text{TP}}. \tag{8}$$

Similarly, Score_{FN} corresponds to reference edits that the GEC system missed, calculated as the sum of mass that was not transported to the reference edits:

$$\text{Score}_{\text{FN}} = \sum_{j} b_j - \text{Score}_{\text{TP}}. \tag{9}$$

Given Score_{TP}, Score_{FP}, and Score_{FN}, we calculate precision, recall, and F_β:

$$\text{Precision} = \frac{\text{Score}_{\text{TP}}}{\text{Score}_{\text{TP}} + \text{Score}_{\text{FP}}}, \tag{10}$$

$$\text{Recall} = \frac{\text{Score}_{\text{TP}}}{\text{Score}_{\text{TP}} + \text{Score}_{\text{FN}}}. \tag{11}$$

$$F_\beta = \frac{\left(1+\beta^2\right) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}. \tag{12}$$

Following existing edit-level metrics (Bryant et al., 2017; Gong et al., 2022), we use $\beta = 0.5$ as the final sentence-level score. When multiple references are available, the reference that yields the highest $F_{0.5}$ is selected for each input sentence, following existing reference-based metrics (Bryant et al., 2017; Gong et al., 2022; Koyama et al., 2024).

4 Experiments

4.1 Meta-evaluation Settings

Meta-Evaluation Dataset. Our proposed method is evaluated using the meta-evaluation benchmark SEEDA (Kobayashi et al., 2024b) and GMEG-Data (Napoles et al., 2019). SEEDA is a dataset that includes the outputs of 14 systems, including newer models such as GPT-3.5 (Brown et al., 2020), and their human evaluation rankings. It supports two types of domains according to the correction tendencies of the systems: the Base setting that ranks 12 systems excluding GPT-3.5 outputs and fluent reference sentences, and the +Fluency setting that ranks all 14 systems including them. There exist two variants for each setting: SEEDA-E, based on edit-level human evaluation, and SEEDA-S, based on sentence-level human evaluation. Kobayashi et al. (2024b) reported the importance of matching the granularity of evaluation between human and automatic evaluations. Since UOT-ERRANT is an edit-level metric, we evaluate it on SEEDA-E. GMEG-Data is a meta-evaluation dataset that incorporates writing by native speakers. While SEEDA and GJG15 target writing by language learners, GMEG-Data enables meta-evaluation in scenarios with fewer errors. In this study, we use the Wiki domain, which covers the formal style written by native speakers. The outputs to be evaluated consist of eight systems: the source, human correction results (reference), and six GEC systems. In the original paper, the human correction results were assessed by sampling from multiple references; however, because the results of that sampling have not been released, we cannot reproduce it. Given that four references are available in GMEG-Data, we treat the first reference as a reference system in our experiments. As shown in Figure 2, the GEC systems in the Wiki domain made fewer edits per sentence than those for SEEDA, facilitating meta-evaluation in situations with lower error density.

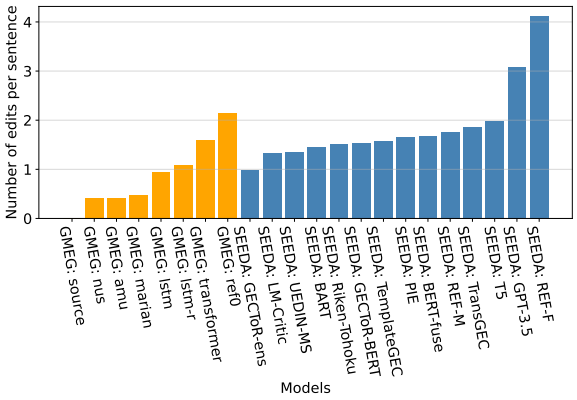


Figure 2: System-wise average number of edits per sentence in both Wiki domain of GMEG-Data (Orange) and SEEDA (Blue). For each dataset, the bars are sorted by the heights.

Computation of System-level Scores. Human evaluation calculates system-level scores by aggregating sentence-level scores. Specifically, it converts sentence-level scores into pairwise comparison results by comparing systems against

each other, then transforms them into system-level scores using rating algorithms such as TrueSkill (Herbrich et al., 2006) or Expected Wins (Bojar et al., 2013). In practice, SEEDA uses TrueSkill-based scores as their final scores. Our implementation uses GEC-METRICS (Goto et al., 2025a)[1] with each system's default settings. Goto et al. (2025b) pointed out that aggregation methods in previous automatic evaluations differed from human evaluations and reported that aligning them improved evaluation performance. Following this insight, our experiments adopt TrueSkill-based aggregation for all metrics.

Reference Set. While SEEDA provides source, hypotheses, and human evaluations for the hypotheses, it does not provide references. To assess the robustness of the metrics, we conduct meta-evaluation with diverse reference sets, following Ye et al. (2023). **Official** refers to the two official references used in the CoNLL-2014 shared task (Ng et al., 2014). **10 Refs** consists of 10 unofficial references annotated by Bryant and Ng (2015). **E-Minimal**, **NE-Minimal**, **E-Fluency**, and **NE-Fluency** are annotated by Sakaguchi et al. (2016). These are combinations of the annotator's expertise, expert (E-) and non-expert (NE-), and domains regarding the purpose of GEC, minimal edit (-Minimal), and fluency edit (-Fluency). For the SEEDA-E Base, we use four types: Official, 10refs, E-Minimal, and NE-Minimal, and for +Fluency, we use E-Fluency and NE-Fluency. Note that, since one of the E-Minimal and one of the E-Fluency are already used as SEEDA systems, they are excluded and used as a single reference, respectively. For GMEG-Data, although it provides four references, we use the second, third, and fourth references as a reference set because we use the first reference as a GEC system to be evaluated.

UOT-ERRANT Settings. We use mean pooling representation of ELECTRA base [2] (Clark et al., 2020) for $\mathrm{Enc}(\cdot)$ in Equation 1. We use the stabilized algorithm (Schmitzer, 2019) implemented in the POT library (Flamary et al., 2021, 2024) to solve the UOT problem. The calculation of Equation 4 requires three hyperparameters: ϵ, λ_1, and λ_2. Since direct parameters tuning on SEEDA is not possible due to test leakage, we tune these parameters using GJG15 (Grundkiewicz et al., 2015) as development data. GJG15 comprises the 12 system outputs submitted to CoNLL-2014 shared task (Ng et al., 2014) and their human evaluation. As there is no need to uniformize the transport plan, we fix the weight of the entropy regularization term ϵ to 0.1, following Arase et al. (2023). We assume that $\lambda_1 = \lambda_2$ and vary them in the range [0.02, 1.00] with a step of 0.02. We use the parameters that yield the highest Pearson correlation with human evaluations in GJG15. When the Pearson correlation is the same, we use parameters with the higher Spearman correlation. Finally, we use $\lambda_1 = \lambda_2 = 0.1$.

4.2 Existing Metrics to be Compared

ERRANT (Felice et al., 2016; Bryant et al., 2017) is a representative edit-level metric in GEC. Similar to Section 3.2.2, after extracting the edits $\mathcal{E}_{\text{hyp}}$ and $\mathcal{E}_{\text{ref}}$, we calculate precision, recall, and $F_{0.5}$ from their overlap, $\mathcal{E}_{\text{overlap}} = \mathcal{E}_{\text{hyp}} \cap \mathcal{E}_{\text{ref}}$:

$$\text{Prec.} = \frac{\sum_{e \in \mathcal{E}_{\text{overlap}}} w_e}{\sum_{e \in \mathcal{E}_{\text{hyp}}} w_e}, \text{Recall} = \frac{\sum_{e \in \mathcal{E}_{\text{overlap}}} w_e}{\sum_{e \in \mathcal{E}_{\text{ref}}} w_e}. \tag{13}$$

The w_e is the weight of an edit e, and ERRANT always uses $w_e = 1$.

PT-ERRANT (Gong et al., 2022) is based on the same formulation as Equation 13, but calculates w_e using BERTScore (Zhang et al., 2020). Specifically, it uses the absolute value of the difference in scores when applying the edit to the input text as the weight: $w_e = |\text{BERTScore}(S, R) - \text{BERTScore}(S_{\{e\}}, R)|$. We use `bert-based-uncased` [3] for calculating BERTScore.

GLEU (Napoles et al., 2015, 2016) and GREEN (Koyama et al., 2024) are reference-based n-gram metrics. GLEU is a precision-based metric, while GREEN enables the calculation of F_β. For both metrics, we use n ranging from 1 to 4. Following Koyama et al. (2024), we use $F_{2.0}$ in GREEN.

CLEME (Ye et al., 2023) compares the chunk sequences constructed from the edits between the hypothesis and reference. We use CLEME-independent, and the scale factors and thresholds for TP, FP, and FN follow Ye et al. (2023).

[1] https://pypi.org/project/gec-metrics/
[2] google/electra-base-discriminator
[3] google-bert/bert-base-uncased

	↑ SEEDA-E Base Setting								↑ SEEDA-E +Fluency Setting				↑ GMEG-Data		
	Official		10 Refs		E-Minimal		NE-Minimal		E-Fluency		NE-Fluency				
Metrics	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	↓ Avg. Rank
n-gram level and reference-based metrics															
GLEU	.909	.965	.949	.958	.848	.916	.808	.895	.278	.600	**.781**	**.921**	.785	.619	3.214
GREEN	.912	.965	.910	.979	.858	.930	.700	.825	**.547**	**.802**	.745	.908	**.871**	**.786**	2.786
Edit-level and reference-based metrics															
ERRANT	.881	.895	.952	.951	.864	.804	.740	.720	-.005	.424	.114	.508	.780	.667	4.857
PT-ERRANT	.924	.951	**.957**	.986	.894	.888	.802	.832	.005	.310	.276	.578	.830	.690	3.286
CLEME	.910	.930	.932	.937	**.910**	**.965**	.716	.706	-.043	.297	.473	.653	.825	.667	4.286
UOT-ERRANT	**.950**	**.979**	.942	**.993**	.901	.909	**.915**	**.930**	.445	.684	.705	.851	.699	.690	**2.357**
Sentence-level and reference-free metrics															
SOME	.893 / .944								.965 / .965				.818	.548	-
Scribendi	.837 / .888								.826 / .912				.822	.571	-
IMPARA	.901 / .944								.969 / .965				.868	.619	-
Qwen3-8B-S	.886 / .965								.418 / .939				.869	.976	-
GPT-4-E♠	.905 / .986								.848 / .987				N/A	N/A	-

Table 1: System-level meta-evaluation results on SEEDA-E, with various reference sets. r is Pearson correlation and ρ is Spearman correlation. **Bold** represents the highest value in each column, and underline represents the second highest. "Avg. Rank" refers to the average rank when the results in each column are converted into rankings, indicating the robustness of reference-based metrics to the reference. The symbol ♠ denotes that we cite the values reported in the original paper (Kobayashi et al., 2024a).

SOME (Yoshimura et al., 2020) is a reference-free metric based on a weighted sum of scores for grammaticality, fluency, and meaning preservation. We use the official pre-trained weights [4], and 0.55, 0.43, and 0.02 as the weights for grammaticality, fluency, and meaning preservation, respectively. These are the weights tuned for sentence-level evaluation by Yoshimura et al. (2020).

Scribendi (Islam and Magnani, 2021) evaluates hypotheses by confirming that the perplexity calculated by the language model has decreased and that surface consistency has been maintained. GPT-2 (Radford et al., 2019) is used as the language model, and 0.8 is used as the threshold for surface consistency.

IMPARA (Maeda et al., 2022) uses a similarity model to confirm that the meaning has been preserved by the correction, and then calculates the score using a quality estimation model. We use `bert-base-cased` as the similarity model, the unofficial quality estimation model published by Goto et al. (2025a) [5], and a similarity threshold of 0.9.

[4] https://github.com/kokeman/SOME

[5] https://huggingface.co/gotutiyan/IMPARA-QE. Note that no official pre-trained models are available.

LLM-S and LLM-E (Kobayashi et al., 2024a) are an LLM-as-a-judge-based metric that takes multiple hypotheses and estimates a 5-level score for each of the hypotheses. The difference between LLM-S and LLM-E lies in whether the hypothesis is input as a sentence or converted into an editing sequence before input. Goto et al. (2025a) mentions that they could not fully reproduce the results reported in the original paper, particularly for LLM-E. Thus, we cite the values of GPT-4-E reported in Kobayashi et al. (2024a), when using GPT-4 (OpenAI et al., 2024) as a LLM, to avoid underestimating this metric. Furthermore, since Kobayashi et al. (2024a) did not report results for GMEG-Data, we also report our LLM-S experimental results based on Qwen3-8B (Yang et al., 2025), denoted as Qwen3-8B-S.

4.3 Experimental Results

Table 1 shows the correlations with human evaluation in SEEDA-E. The proposed method achieved superior performance over existing metrics across various reference sets. In Base, the proposed method often ranks first, and in +Fluency, it achieved the highest correlation among edit-level metrics. +Fluency evaluates performance in situations where more edits occur, leading to a wider

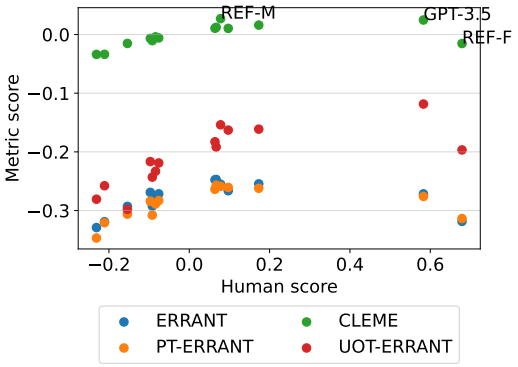


Figure 3: Scatter plot for 14 systems on SEEDA-E +Fluency. The x-axis and y-axis represent human and metric scores, respectively.

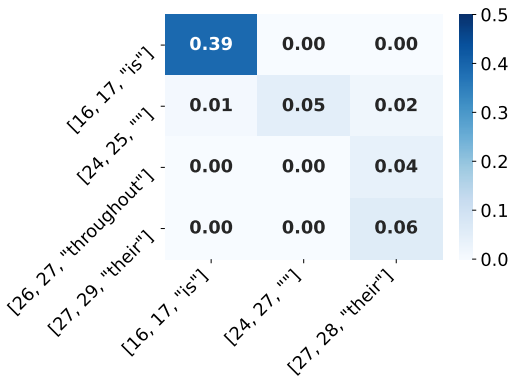


Figure 4: A case study: Alignment between four hypothesis edits (y-axis) and three reference edits (x-axis). The actual sentences are as follows:
Source: "It is still early for parents to decide whether they can foster a new life that are not able to work and may suffer the pain in the entire life ."
Reference: "... new life that is not able to work and may suffer their entire life ."
Hypothesis: "... new life that is not able to work and may suffer pain throughout their life ."

variety of edit patterns compared to Base. In such scenarios, it is challenging to capture diverse correct edits with a small number of references (Choshen and Abend, 2018; Bryant and Ng, 2015). Our method solves this problem by considering edit similarity via edit vector. "Avg. Rank" in Table 1 shows the average rank of the correlations in each column for reference-based metrics. Our method achieved the highest average rank among the reference-based metrics, indicating that it enables robust evaluation across diverse domains and reference sets.

In GMEG-Data, UOT-ERRANT showed the second-highest Spearman correlation among reference-based metrics, but the lowest Pearson correlation. This result is likely due to differences in correction tendencies within the dataset. GMEG-Data contains more punctuation edits compared to SEEDA. Punctuation edits account for 32.1% (2,241 / 6,976) of the total edits across all systems, which is in contrast to SEEDA's 5.9% (599 / 10,062). Since edit vectors are intended to capture the semantic similarity of edits, they are less effective for punctuation edits, making them unsuitable for evaluating GMEG-Data. Therefore, the proposed method is an effective metric in situations where the surface forms of edits are diverse, such as in SEEDA's +Fluency.

To investigate the factors behind the improved correlation in +Fluency, we present a scatter plot in Figure 3. The x-axis and y-axis represent the human evaluation scores and metric scores, respectively, for 14 systems with the representative system labels of *REF-M*, *GPT-3.5*, and *REF-F*. REF-M includes a minimal correction, while GPT-3.5 and REF-F include relatively more edits. The ideal plot would show a direct alignment on the diagonal line from the bottom-left to the top-right. The difference between SEEDA's Base and +Fluency lies in whether GPT-3.5 and REF-F are included in the system set. Therefore, the performance improvement in +Fluency should be due to improved evaluation of these two systems. The results show that UOT-ERRANT is able to assign relatively high scores to the two systems. ERRANT, PT-ERRANT, and CLEME assign higher scores to REF-M than to GPT-3.5, leading to lower correlation.

UOT-ERRANT also offers high explainability by visualizing the alignment between edits. Figure 4 shows a transport plan using an actual example. This transport plan can be interpreted as an alignment that explains how the hypothesis edits match the reference edits. In this example, existing metrics based on strict matching, such as ERRANT, only recognize the hypothesis edit [16, 17, "is"] as correct, while evaluating the other three hypothesis edits as incorrect. In contrast, UOT-ERRANT assigns a partial score by aligning the hypothesis edit [24, 25, ""] with the reference edit [24, 27, ""] and [27, 29, "their"] with [27, 28, "their"]. Intuitively, these edits are not completely incorrect, and UOT-ERRANT metric captures this

	Base Official		+Fluency NE-Fluency	
	r	ρ	r	ρ
Default settings (same as Table 1)				
UOT-ERRANT	.950	.979	.705	.851
Vectorization (Default: By removing an edit))				
By adding an edit	.956	.979	.430	.657
Mass (Default: L^2 norm)				
Uniform	.932	.944	.678	.815
Cost (Default: Euclidean distance)				
Cosine similarity	.920	.937	.453	.714
Enc$(\cdot)$ (Default: ELECTRA-base)				
RoBERTa-base	.959	.916	.315	.543
LaBSE	.893	.930	.879	.960

Table 2: Ablation study on SEEDA-E. The *UOT-ERRANT* on the top line indicates the results with our default settings, which is the same results as Table 1. The following lines show results when replacing the default setting with others.

intuition by flexible edit-level evaluation based on the transport plan.

Although Table 1 confirms that UOT-ERRANT improves correlation, especially under +Fluency, its correlation remains lower compared to n-gram-level and sentence-level metrics. However, the actual role of a metric is not only to rank systems, but also to analyze weaknesses and strengths of GEC systems. Correlations with human evaluation commonly used in meta-evaluation reflect only ranking performance and are not suitable for measuring analytical capabilities. Such analytical functionality is limited to edit-level metrics, and UOT-ERRANT achieves the highest evaluation performance among them. In this sense, UOT-ERRANT offers the advantage of enabling analysis based on precision, recall, F_β, and edit alignments, which is shown in Figure 4.

4.4 Ablation Studies

We conduct ablation studies to verify how the hyperparameters of UOT-ERRANT improve GEC evaluation. We measure the meta-evaluation performance when varying hyperparameters under two configurations: Base with Official references, and +Fluency with NE-Fluency references. All results regarding UOT is based on the experimental setting described in Section 4.1.

Vectorization Edit vectors quantify the semantic impact of edits by removing them from a corrected sentence, but they can also be quantified by adding edits to the source, similar to PT-ERRANT's weight calculation. The difference between the two is whether the vectors are contextualized by other edits within the same sentence. In the example in Figure 1, the edit [23, 24, "takes"] is in both the hypothesis and reference edits. In adding an edit, it is quantified as the same vector in both edit sets, but in removing edit, surrounding edits such as [27, 28, "given"] in the reference edits also influence the vector. The results in *By adding an edit* of Table 2 show that removing an edit is better than adding it for vectorization on +Fluency. This suggests that the edit vectors contexualized by surrounding edits accurately quantifies the edit's impact for semantic changes.

Mass We use uniform weights instead of the default L^2 norm, in order to measure the contribution of our proposed edit weighting as mentioned in Section 3.2.2. The correlation values in *Uniform* in Table 2 show that using the L^2 norm is better than the uniform. This result is consistent with previous research that weighting edits improves correlation, as reported in PT-ERRANT (Gong et al., 2022) and CLEME (Ye et al., 2023).

Cost The default setting is Euclidean distance, but we use cosine similarity instead to quantify the vector's length in the cost. The correlation values in *Cosine similarity* in Table 2 shows that Euclidean distance is better than cosine similarity, indicating that the vector's length is important for the cost definition.

Sentence Encoder We use RoBERTa-base (Liu et al., 2019) and LaBSE (Feng et al., 2022) instead of ELECTRA-base as a sentence encoder, Enc$(\cdot)$ in Equation 1. We use the mean pooling representation for RoBERTa, and `[CLS]` representation for LaBSE, considering pre-trained setting. The results are shown in the most bottom block in Table 2. Drop for RoBERTa-base implies the importance of the strong signals for each edit in the encoder. The gains in LaBSE in +Fluency reflect the importance of the interactions of edits in the corrected sentence, but the drops in Base shows its weakness to quantify a single edit.

Error Type	L^2 Norm (Mean $\pm$ Stdev.)
ORTH	0.170 $\pm$ 0.406
PUNCT	0.835 $\pm$ 0.597
ADV	0.944 $\pm$ 0.485
ADJ	0.962 $\pm$ 0.702
NOUN:NUM	0.977 $\pm$ 0.593
MORPH	0.977 $\pm$ 0.577
PREP	1.033 $\pm$ 0.477
VERB:FORM	1.066 $\pm$ 0.473
DET	1.084 $\pm$ 0.558
VERB:TENSE	1.087 $\pm$ 0.510
NOUN	1.109 $\pm$ 0.633
VERB	1.171 $\pm$ 0.666
VERB:SVA	1.231 $\pm$ 0.654
PRON	1.247 $\pm$ 0.683
CONJ	1.247 $\pm$ 0.556
OTHER	1.318 $\pm$ 0.685
SPELL	1.409 $\pm$ 0.610

Table 3: The mean and standard deviation of the L^2 norm of edit vectors for each error type.

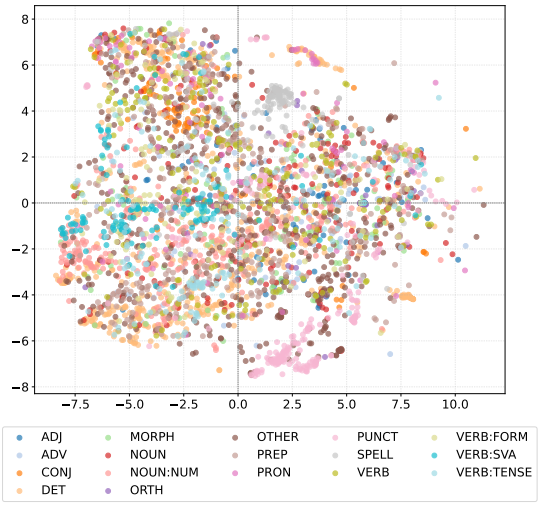


Figure 5: Visualization of edit vectors with dimensionality reduced by t-SNE. The plot is color-coded according to the error type.

5 Discussion

5.1 Characteristics of Edit Vector

To analyze the properties of edit vectors, we analyze the 3,859 edits in SEEDA's GPT-3.5 output.

5.1.1 Norm Varies Depending on Error Types

The norm of the edit vector represents how strongly the edit changes the semantics of a sentence. To investigate the relationship between the qualitative properties of edits and their norms, we focus on the error type of the edits. The error type is a classification system that describes the nature of the edit, primarily based on parts of speech. Examples of these classifications include SPELL, NOUN:NUM (Noun number), and DET (Determiner). Here, we use the error types proposed by ERRANT (Bryant et al., 2017).

Table 3 shows the mean and standard deviation of the edit vector norms for error types with a frequency of 50 or more. It is clear that the norm varies significantly depending on the error type. For instance, edits related to ORTH (Orthography, e.g., case and/or whitespace) and PUNCT (Punctuation) have smaller norms, while edits concerning content words such as VERB and NOUN have larger norms. Since edits to content words cause obvious changes in meaning, this tendency coordinates with our intuition. In the context of UOT-ERRANT, the norm acts as an implicit weight for the edits, which have a direct contribution to the improvement in evaluation performance.

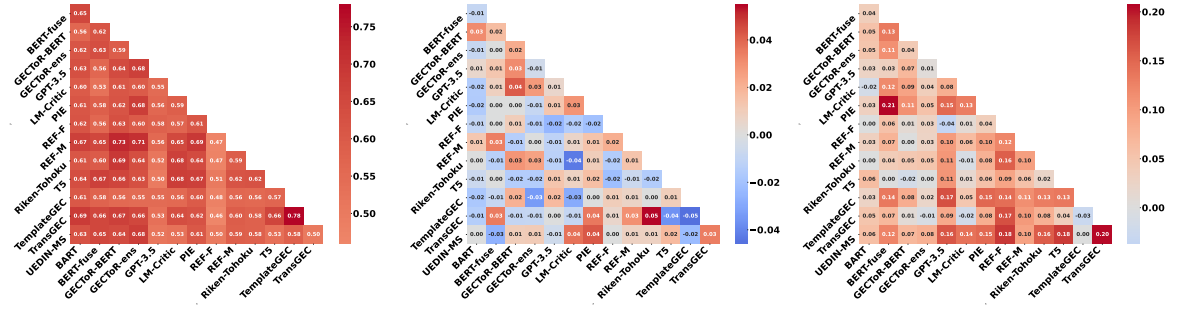
5.1.2 Similar Edits Become Similar Vectors

To verify that similar edits are encoded in similar spaces, we use t-SNE (van der Maaten and Hinton, 2008) to reduce dimensionality and visualize the vectors for each error type. Figure 5 shows the plot for error types with a frequency of 50 or more. We observed that edits for PUNCT, SPELL, and CONJ (Conjunction) tend to form clusters. Furthermore, NOUN:NUM and VERB:SVA (Subject-verb agreement) are also positioned in proximity in the global view. As these two error types often share the commonality of adding an "s" to the end of a word, this suggests that the edit vector partially considers surface-level changes.

In contrast, edits such as NOUN, VERB, and ADJ (Adjective) are encoded in a scattered space. For these edits, the direction of semantic change varies depending on the vocabulary, thus the vectors within the error type group are not consistent. This result rather emphasizes that the edit vectors appropriately capture semantic changes.

5.2 Accuracy of Pairwise Comparison

Section 4.3 showed that UOT-ERRANT improves the correlation in +Fluency by assigning higher credits to systems such as GPT-3.5 and REF-F. Given that we convert sentence-level pairwise comparison scores into system rankings using TrueSkill, this improvement comes



(a) ERRANT (b) Improvement of PT-ERRANT (c) Improvement of UOT-ERRANT

Figure 6: Sentence-level meta-evaluation results broken down by system pairs. Each cell shows the agreement rate between human and automatic evaluation in pairwise comparisons. ERRANT shows the absolute values, while PT-ERRANT and UOT-ERRANT indicate the improvement over ERRANT.

from better pairwise comparisons. To analyze this, we extend sentence-level meta-evaluation in GEC (Kobayashi et al., 2024b), which calculates the agreement rate of pairwise comparison results between human and automatic evaluations, to compute the agreement rate for each system pair.

Figure 6 illustrates the agreement rate of pairwise comparison results, broken down by system pair. We use +Fluency of SEEDA-E and NE-fluency as references. ERRANT's results (Fig. 6a) are presented as absolute scores, while the agreement rates for PT-ERRANT (Fig. 6b) and UOT-ERRANT (Fig. 6c) are shown as improvement from ERRANT. For ERRANT, the agreement rate tends to drop when pairs include GPT-3.5 or REF-F. Although PT-ERRANT extends ERRANT by adding edit weighting, it shows no meaningful improvement in the agreement rates. In contrast, UOT-ERRANT improves the agreement rate for pairs involving GPT-3.5 or REF-F by over 10% in some cases. This improvement in pairwise comparison directly contributes to the higher correlation observed in Section 4.3.

5.3 Computation Cost

Since the proposed method uses a neural model to calculate edit vectors, its computational cost is higher than that of ERRANT. While this could be a weakness, it is a necessary cost given the interpretability of UOT-ERRANT and the high correlation observed in the +Fluency setting of SEEDA-E. To experimentally investigate the actual computational cost, we measure the time required to evaluate each of the 13 systems used in the +Fluency setting of SEEDA. Each output has 391 sentences, but the number of edits is different across systems.

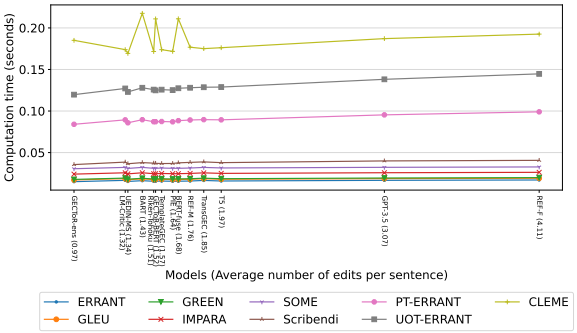


Figure 7: Actual average computation time per sentence (seconds). The horizontal axis shows system names and the average number of edits per sentence. The systems are plotted with intervals corresponding to the number of edits per sentence which is shown in the parentheses.

In particular, we focus on the computation time per system to examine how the cost varies relative to the number of edits. This is also intended to verify whether larger-scale experiments will remain feasible when dealing with domains that contain a higher number of edits in the future. We use the metrics used in Table 1, but we exclude LLM-S and LLM-E because they scores multiple system outputs at once, thus we cannot compute system-wise computation time. All experiments were conducted on a single NVIDIA RTX 3090.

The results are shown in Figure 7. UOT-ERRANT tends to have a higher computational cost compared to other metrics. First, compared to ERRANT, GLEU, and GREEN, UOT-ERRANT demands approximately eight times more time because it requires neural model computations. Second, while SOME, IMPARA, and Scribendi also require neural model computations, they are

sentence-level metrics, meaning that the number of forward passes equal to the number of sentences. In contrast, UOT-ERRANT requires a number of passes proportional to the number of edits. Actually, the computation time tends to increase when evaluating systems with a large number of edits, such as GPT-3.5 or REF-F. Next, when compared with PT-ERRANT, UOT-ERRANT took about 1.5 times longer. This difference in cost can be explained by the presence of optimal transport calculation. Finally, CLEME exhibited the highest cost due to the implementation problem: inclusion of file I/O from executing multiple scripts in the CLI. Other metrics do not encounter this problem as they can be run in a single script via GEC-METRICS (Goto et al., 2025a). Although these are implementation-level differences, the same issues would arise in actual use cases.

In summary, while UOT-ERRANT requires a relatively high computational cost, we believe it is acceptable considering the improvement in evaluation performance shown in Table 1. Since evaluation is fundamentally a one-time process, the computational cost is not a major problem. It might become an issue in the future if online execution is required, such as when applying it as a reward for reinforcement learning. A potential solution is to accelerate the computation by caching the results of the embedding model or by implementing appropriate batching. For example, if multiple GEC systems output the same corrected sentence, the result for one can be reused for the others.

6 Conclusion

In this study, we propose a new GEC metric based on *edit vectors* to calculate the similarity between a hypothesis and a reference edits. The edit vector is defined as the difference vector between sentence representations before and after removing edits, providing a novel way to quantify edits. Furthermore, we propose UOT-ERRANT, based on the idea of optimally transporting edit vectors from the hypothesis to the reference. Meta-evaluation using the SEEDA dataset showed that UOT-ERRANT improved conventional edit-level metrics, particularly in domains with a high number of edits. Our analysis revealed that edit vectors effectively reflect the qualitative characteristics of edits, demonstrated by analyzing their norms and clusters based on error types.

Our proposed *edit vector* can be applied to a variety of tasks involving partial edits. For example, in text simplification (Shardlow et al., 2024), rewriting a difficult word or phrase into a simpler one can be regarded as an edit. Similarly, in automatic speech recognition error correction (Leng et al., 2021), edits occur much like in GEC, allowing for the calculation of difference vectors. In the vision field, image editing tasks are well-known (Avrahami et al., 2022). Images before and after editing can be encoded separately by an image encoder, and their difference can be taken to vectorize the edit. In the future, we anticipate that applying the edit vector to other tasks will broaden its applications, particularly in evaluation-centric scenarios. The limitations and ethical statements of this study are discussed in Appendices A and B.

Acknowledgments

We thank the action editor and the anonymous reviewers for their valuable comments. This work has been supported by JST SPRING Grant Number JPMJSP2140.

References

Yuki Arase, Han Bao, and Sho Yokoi. 2023. Unbalanced optimal transport for unbalanced word alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3966–3986, Toronto, Canada. Association for Computational Linguistics.

Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218. IEEE.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

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. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. 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)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 697–707, Beijing, China. Association for Computational Linguistics.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. 2017. Scaling algorithms for unbalanced transport problems.

Leshem Choshen and Omri Abend. 2018. Inherent biases in reference-based evaluation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Melbourne, Australia. Association for Computational Linguistics.

Julian Chow, Lucia Specia, and Pranava Madhyastha. 2019. WMDO: Fluency-based word mover's distance for machine translation evaluation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 494–500, Florence, Italy. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pierre Colombo, Guillaume Staerman, Chloé Clavel, and Pablo Piantanida. 2021. Automatic text evaluation through the lens of Wasserstein barycenters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10450–10466, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. 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*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. 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*, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. 2021. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Rémi Flamary, Cédric Vincent-Cuaz, Nicolas Courty, Alexandre Gramfort, Oleksii Kachaiev, Huy Quang Tran, Laurène David, Clément Bonet, Nathan Cassereau, Théo Gnassounou, Eloi Tanguy, Julie Delon, Antoine Collas, Sonia Mazelet, Laetitia Chapel, Tanguy Kerdoncuff, Xizheng Yu, Matthew Feickert, Paul Krzakala, Tianlin Liu, and Eduardo Fernandes Montesuma. 2024. POT python optimal transport (version 0.9.5).

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. 2015. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6891–6902, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025a. gec-metrics: A unified library for grammatical error correction evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 524–534, Vienna, Austria. Association for Computational Linguistics.

Takumi Goto, Yusuke Sakai, and Taro Watanabe. 2025b. Rethinking evaluation metrics for grammatical error correction: Why use a different evaluation process than human? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1165–1172, Vienna, Austria. Association for Computational Linguistics.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. 2020. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

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

Md Asadul Islam and Enrico Magnani. 2021. Is this the end of the gold standard? A straightforward reference-less grammatical error correction metric. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3009–3015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leonid V Kantorovich. 1958. On the translocation of masses. *Manage. Sci.*, 5(1):1–4.

Anisia Katinskaia and Roman Yangarber. 2024. GPT-3.5 for grammatical error correction. In

Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 7831–7843, Torino, Italia. ELRA and ICCL.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024a. 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)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024b. Revisiting meta-evaluation for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 12:837–855.

Shota Koyama, Ryo Nagata, Hiroya Takamura, and Naoaki Okazaki. 2024. n-gram F-score for evaluating grammatical error correction. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 303–313, Tokyo, Japan. Association for Computational Linguistics.

Yuto Kuroda, Atsushi Fujita, and Tomoyuki Kajiwara. 2024. Word-level translation quality estimation based on optimal transport. In *Proceedings of the 16th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 209–224, Chicago, USA. Association for Machine Translation in the Americas.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Yichong Leng, Xu Tan, Rui Wang, Linchen Zhu, Jin Xu, Wenjie Liu, Linquan Liu, Xiang-Yang Li, Tao Qin, Edward Lin, and Tie-Yan Liu. 2021. FastCorrect 2: Fast error correction on multiple candidates for automatic speech recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4328–4337, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Koki Maeda, Masahiro Kaneko, and Naoaki Okazaki. 2022. IMPARA: Impact-based metric for GEC using parallel data. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3578–3588, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. ArXiv preprint arXiv:1301.3781.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. 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 (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. GLEU without tuning. ArXiv preprint arXiv:1605.02592.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and

Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla

Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. ArXiv preprint arXiv:2303.08774.

Ofir Pele and Michael Werman. 2009. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Bernhard Schmitzer. 2019. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024. The BEA 2024 shared task on the multilingual lexical simplification pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.

Richard Sinkhorn. 1964. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879.

Kyle Swanson, Lili Yu, and Tao Lei. 2020. Rationalizing text matching: Learning sparse alignments via optimal transport. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5609–5626, Online. Association for Computational Linguistics.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin

Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Jingheng Ye, Yinghui Li, Qingyu Zhou, Yangning Li, Shirong Ma, Hai-Tao Zheng, and Ying Shen. 2023. CLEME: Debiasing multi-reference evaluation for grammatical error correction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6174–6189, Singapore. Association for Computational Linguistics.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mikhail Yurochkin, Sebastian Claici, Edward Chien, Farzaneh Mirzazadeh, and Justin M Solomon. 2019. Hierarchical optimal transport for document representation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Appendix

A Limitations

Meta-evaluation This paper conducted meta-evaluation using SEEDA and confirms an improved correlation with edit-level human evaluation. This meta-evaluation includes both Base and +Fluency domains, assumed to cover general use cases. However, for other cases, it's also possible to consider domains with low error density, as proposed in CWEB (Flachs et al., 2020), or the correction of texts by writers not limited to learners, as discussed in GMEG-data (Napoles et al., 2019). Current meta-evaluation relying on SEEDA has limitations in such domains, but we look forward to future extensions of meta-evaluation datasets.

Encoder Models Since edit vectors are based on sentence representations, the quality of the vectors depends on the quality of the encoder model. In this study, we adopted ELECTRA because its pre-training tasks are similar to grammatical error detection, but other models might show better evaluation performance. As shown in Section 2, the optimal encoder varies depending on the target domain of evaluation. While we could explore the optimal encoder in this study, it would merely reproduce overfitting to the specific benchmark. Given that users practically need to find the best model for their specific domain, we believe there is little value in discussing encoder selection in depth here. Furthermore, it is also interesting to investigate which model characteristics lead to higher-quality editing vectors. While model probing may be useful for this purpose, the methodology for investigating its relationship with editing vectors has not yet been fully established and remains it for future work.

Computation Cost As stated in Section 5.3, the proposed method requires more computation time than other metrics because it involves neural model computations for each edit. However, as shown in Figure 7, the computation time per sentence is less than 0.15 seconds, which we consider to be sufficiently fast. Furthermore, since a single execution is typically sufficient for evaluation, the computational cost does not become a major problem in most cases.

B Ethical Statements

Social Bias The proposed method implicitly weights edits by using the norm of the edit vectors. As shown in the ablation study (§4.4), while this weighting is important, it can also lead to bias. For example, if the weight of an edit changes based on different social attributes like gender or nationality, there's a risk that the metric could favor certain attributes. However, this issue is not exclusive to UOT-ERRANT; it could also exist in PT-ERRANT and LLM-based metrics. We would like to defer a comprehensive discussion on this point to future research.