

Time Is Effort: Estimating Human Post-Editing Time for Grammar Error Correction Tool Evaluation

Ankit Vadehra*

University of Waterloo, Vector Institute
avadehra@uwaterloo.ca

Bill Johnson

Scribendi Inc.
bill.johnson@scribendi.com

Gene Saunders

Scribendi Inc.
gene.saunders@scribendi.com

Pascal Poupart

University of Waterloo, Vector Institute
ppoupart@uwaterloo.ca

Abstract

Text editing can involve several iterations of revision. Incorporating an efficient Grammar Error Correction (GEC) tool in the initial correction round can significantly impact further human editing effort and final text quality. This raises an interesting question to quantify GEC Tool usability: How much effort can the GEC Tool save users? We present the first large-scale dataset of post-editing (PE) time annotations and corrections for two English GEC test datasets (BEA19 and CoNLL14). We introduce Post-Editing Effort in Time (PEET) for GEC Tools as a human-focused evaluation scorer to rank any GEC Tool by estimating PE time-to-correct. Using our dataset, we quantify the amount of time saved by GEC Tools in text editing. Analyzing the edit type indicated that determining whether a sentence needs correction and edits like paraphrasing and punctuation changes had the greatest impact on PE time. Finally, comparison with human rankings shows that PEET correlates well with technical effort judgment, providing a new human-centric direction for evaluating GEC tool usability.¹

1 Introduction

Grammar Error Correction (GEC) is an important step of the text editing process. There has been a lot of work to build automated GEC tools that can improve the structure and fluency of text while also correcting language errors (Bryant et al., 2023). Since GEC tool-assisted text editing is an iterative process, an editor can make post-edits to the tool output to obtain the closest targeted correction. Estimating the post-editing (PE) effort required to reach the targeted correction can be used as a quality evaluation for the tool.

Human-in-the-loop PE effort was introduced and explored extensively for Machine Translation (MT) (Koponen, 2016) systems. PE effort is studied across three levels (Kittredge, 2002): technical effort, which is the number of edits; cognitive effort, which denotes the psychological assessment required to identify and correct the errors; and temporal effort, which is the total time taken to evaluate and perform post-edits (which includes technical and cognitive effort). Ye et al. (2021) and Tezcan et al. (2019) have explored estimating MT PE time based on edit features. Technical PE effort has also been studied in areas like Text Summarization (Lai et al., 2022), Natural Language Generation (Sripada et al., 2005) and GEC (Rozovskaya and Roth, 2021; Östling et al., 2024).

To incorporate the human editor effort in text correction, we present the first work to consider PE effort in Time (PEET) scores for quality estimation of a GEC tool. The usability of a GEC tool depends inversely on the PE effort to fix the tool output. We release the first large-scale dataset capturing time-to-correct annotations for two English GEC test sets - BEA19 (Bryant et al., 2019) and CoNLL14 (Ng et al., 2014), post-edited from two conditions: the original sentence and the output from two strong GEC tools - GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019). We further present a new human-centric GEC Tool evaluation method - PEET Scorer, to estimate the time-to-correct for GEC Tool predictions, which correlates well with human editing effort. As a result, we propose that the PEET scorer can be incorporated along with Post-Editing to evaluate a GEC Tool from a human editor’s perspective.

In this work, we make the following contributions:

1. We present the first large-scale GEC dataset with post-editing time-to-correct annotations along with three new high-quality human-

*Corresponding author.

¹We release our dataset and code at - https://github.com/ankitvad/PEET_Scorer

preference targeted correction sets for two GEC Test datasets (BEA19 and CONLL14) - source sentence correction and post-edit for two strong GEC Tools (GECToR and GEC-PD) output.

2. We quantify the editing time saved and improvement in final correction quality (estimated using GEC metrics) using GEC Tools for first-pass text-editing. We also observe that determining whether a sentence needs correction and edits like paraphrasing and punctuation changes has the greatest impact on time-to-correct.
3. We contribute a new evaluation method called PEET Scorer that can be used to rank any GEC Tool in terms of time-to-correct. We compare the PEET scorer with 3 human judgment rankings of 33 GEC Tools, and demonstrate high correlation with further correction effort required.

2 Background Work

2.1 Grammar Error Correction (GEC) Tools

GEC tools can be broadly divided into supervised-trained, LLM-based, and ensemble-ranked models (Omelianchuk et al., 2024).

The supervised GEC tools can be divided into edit-based and sequence-to-sequence models. Edit-based models convert the task to a sequence-tagging and editing approach where each token in the input sentence is assigned an edit operation. Some tools that use this approach are the PIE (Awasthi et al., 2019) and GECToR (Omelianchuk et al., 2020; Tarnavskyi et al., 2022) models. Sequence-to-Sequence (S2S) GEC Tools utilize an encoder-decoder architecture where the corrected sentence is generated for each input sentence (Choe et al., 2019; Grundkiewicz et al., 2019; Kiyono et al., 2019).

Large language models like Llama (Touvron et al., 2023; Omelianchuk et al., 2024) and ChatGPT (Katinskaia and Yangarber, 2024) also perform well for GEC (Zhang et al., 2023; Fang et al., 2023b) in different settings like - Zero-Shot, Few-Shot and Fine-Tuning (Korniienko, 2024; Davis et al., 2024; Raheja et al., 2023). The current state-of-the-art GEC tools all rely on the approach of ensembling multiple strong GEC Tools, aggregating them with methods like majority votes (Tarnavskyi

et al., 2022) and logistic regression (Qorib and Ng, 2023; Qorib et al., 2022).

In this work, we use two supervised GEC tools for first-pass text editing: GECToR edit tagging (Omelianchuk et al., 2020) and GEC-PseudoData (GEC-PD) (Kiyono et al., 2019) model, which was trained on a large synthetic corpus. The output of these models is further corrected by human editors while tracking the time-to-correct (temporal effort). We use this time dataset to quantify the impact of GEC tools for text-editing, observing reduced post-editing time and better quality final correction (Section 3.5). Even though the GEC Tools we selected (GECToR and GEC-PD) are not the most recent, they are on par with human-level performance as demonstrated in Section 3.4 - Table 3.

2.2 Post Editing Effort in Machine Translation

Post Editing Effort (PEE) for Quality Estimation is an actively researched task in Machine Translation (MT). It evaluates the output of an MT system for quality and correctness (Senez, 1998; Specia, 2011). Post-editing (PE) the output of an MT system can improve the final translation quality compared to translating the source from scratch, while improving overall editor productivity (Plitt and Masselot, 2010; Guerberof, 2009; Green et al., 2013). We briefly review previous work in MT that explores PEE across three levels (technical, cognitive and temporal effort) (Kittredge, 2002).

Technical effort has been defined by edit distance metrics like - Translation Edit Rate (TER) and Human TER (Snover et al., 2006) as well as keystroke and edit operation logging (Barrachina et al., 2009; O'Brien, 2005; Carl et al., 2011). Cognitive effort has also been studied in terms of edit complexities (Temnikova, 2010; Koponen et al., 2012; Popović et al., 2014; Daems et al., 2017) and human-assessed quality judgment and ranking (Specia et al., 2009, 2011; Koponen, 2012). Keystroke logs to determine pause information (O'Brien, 2005; Carl et al., 2011), eye gaze tracking and pause fixation (Vieira, 2014; Hvelplund, 2014; Daems et al., 2015) and Thinking Aloud Protocol (TAP) (Kittredge, 2002; Vieira, 2017; O'Brien, 2005) have also been proposed as measures of cognitive effort. The work on Temporal Effort in MT estimates the relationship between the time-to-correct and different evaluation metrics (Tatsumi, 2009), source/target translation charac-

teristics (Tatsumi and Roturier, 2010), and quality estimation (Specia, 2011). Zaretskaya et al. (2016) and Popović et al. (2014) study the average temporal effort required for each error type by considering the time-to-correct and frequency of error edits. Finally, Ye et al. (2021) and Tezcan et al. (2019) train models to estimate the post-editing time based on PE features.

PE has also been explored in tasks like Text Summarization (Lai et al., 2022) and Cognitive and Technical PE Effort has been studied for Grammar Error Correction (GEC) evaluation.

2.3 Post Editing Effort in Grammar Error Correction

We review previous work in GEC that closely relates to post-editing (PE) effort across two levels (cognitive and technical effort). To the best of our knowledge, temporal effort for PE has not been explored for GEC tools.

2.3.1 Cognitive Post Editing Effort

Although cognitive PE effort has not been measured directly for GEC, Human judgment rankings of GEC Tools (Grundkiewicz et al., 2015; Kobayashi et al., 2024; Napoles et al., 2019), which are an estimate of perceived cognitive effort, have been used extensively for GEC evaluation metric assessment. Reference-based GEC metrics like ERRANT (Bryant et al., 2017), M^2 (Dahlmeier and Ng, 2012), GoToScorer (Gotou et al., 2020), and GLEU (Courtney et al., 2016) and reference-less metrics like PT- M^2 (Gong et al., 2022), Scribendi Score (Islam and Magnani, 2021), SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) designed to estimate GEC Tool quality are trained and evaluated using the GEC human judgment rankings.

However, perceived cognitive effort does not always agree with the actual PE effort and can be subjective. Sentence correction experiments in GEC have shown poor cognitive agreement between editors. Tetreault et al. (2014) and Tetreault and Chodorow (2008) asked 2 native English speakers to insert a preposition into 200 sentences, from which a single preposition was removed, obtaining an agreement score of just 0.7. Rozovskaya and Roth (2010) asked three annotators to evaluate and mark 200 sentences for correctness, showing a poor pairwise agreement between them (0.4, 0.23, 0.16). Finally, there has been some work considering the cognitive proficiency of the user interacting with a

GEC Tool (Nadejde and Tetreault, 2020) and the annotators who create the evaluation references of GEC test sets (Takahashi et al., 2022; Napoles et al., 2017).

Surprisingly, none of the GEC metrics described above have considered using targeted references (target obtained after correcting the GEC Tool output) to estimate the tool usability dependent on human PE effort.

2.3.2 Technical Post Editing Effort

To the best of our knowledge, only two prior studies have explored the impact of PE technical effort on GEC evaluation. Rozovskaya and Roth (2021) introduced targeted references for English and Russian datasets and Östling et al. (2024) utilize PE references to assess Swedish GEC Tools. The studies show that GEC evaluation using untargeted references ignores the human subjectivity involved in text correction. For instance, the SEEDA - human judgment rankings from Kobayashi et al. (2024) compared the correction outputs of GPT3.5, human editors and various Neural GEC Tools. The GPT-3.5 and human corrections were ranked significantly higher and contained nearly two and three times more edits than other corrections. As a result, these high-quality corrections obtain poor evaluation scores when compared against untargeted references. This inconsistency highlights the importance of PE for GEC Tool evaluation, to capture the true technical effort.

Apart from estimating the PE effort, targeted references can also be used for fine-tuning and aligning Large Language Models (LLMs) with human preferences to generate better outputs (Li et al., 2024).

2.3.3 Temporal Post Editing Effort

We introduce the first work to study the Temporal Effort in PE for GEC. Temporal effort described in terms of time-to-correct can efficiently capture the overall PE effort. We present the first large-scale dataset of post-edited corrections along with their temporal effort annotations for two strong GEC tools, GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019), outputs on two English GEC Test sets - CONLL14 (Ng et al., 2014) and BEA19 (Bryant et al., 2019). We also use this dataset to quantify the impact of GEC Tools in Text Editing and the contribution of different edit types to the human post-editing effort. We present PEET Scorer, a regression-based metric, to estimate the

time-to-correct scores, which can be incorporated along with post-editing to evaluate the usability of GEC Tools in a human-centred manner.

3 Dataset Collection and Processing

An important component in this work is the high-quality dataset of post-edit corrections for GEC, along with their time-to-correct (temporal effort) annotations. We partnered with a professional text-editing company - Scribendi Inc.² to collect this data. This section explains our dataset collection, filtering, and quality estimation process.

3.1 Dataset Source

We use source sentences from two popular English GEC test sets - CONLL14 (Ng et al., 2014) and BEA19 (Bryant et al., 2019) ($1312 + 4477 = 5789$ sentences). Each sentence was corrected in three variations: the source and post-editing outputs from Two GEC Tools - GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019) (Section 2.1). Each sentence variation was corrected by 1 out of 8 professional text editors, employed by Scribendi Inc. This resulted in a dataset of $5789 * 3 = 17367$ target corrections along with their time-to-correct scores.

3.2 Editor Correction Framework

The source sentence and GEC Tool output serve as the basis for further editor correction. This follows the real framework for Text Editing, where a GEC Tool output is evaluated for further correction, compared with the original sentence. The editors were given GEC post-editing (PE) instructions (Appendix F-3) and asked to perform minimal edits and avoid rewrites. We used the Qualtrics³ survey tool to collect PE corrections and enabled the "Timing Question" to log time-to-correct for each source sentence. All other metadata logging was disabled.

The 3 variations for each sentence - source, GEC-ToR and GEC-PD output- were given to a different professional editor (in a pool of 8 editors) to eliminate any time-to-correct bias. The task of evaluating 17,367 sentences was performed in batches of 50. The editors were shown the source sentence and the first-pass GEC Tool output (Appendix F-4). The final target correction and time-to-correct were logged for each sentence. For source sentence correction, only the original sentence was presented.

²<https://www.scribendi.com/>

³<https://www.qualtrics.com/>

3.3 Data Filtering

To improve the dataset quality, we perform two stages of data filtering on the 3 target correction sets for each source (17367 sentences initially). In the first stage, we eliminate outliers based on the logged time-to-correct. Snover et al. (2006) showed that editors took between 3-6 minutes for each correction. Considering this and the distribution of the time-to-correct in our dataset, we filter corrections that took more than 250 seconds (17033 sentences remaining). Finally, we merge duplicate corrections from our dataset by averaging the time-to-correct values (14112 sentences dataset). This filtering allows us to retain 81.26% of our dataset that we use as train and test sets (80:20 split) for the Post-Editing Effort in Time (PEET) Scorer.

3.4 Correction Quality

We collect and present three new target corrections for the CONLL14 (Ng et al., 2014) and BEA19 (Bryant et al., 2019) test datasets. The correction for the source and two post-edited target corrections. We evaluate the quality of the target corrections using the official GEC competition metric and the Inter Annotator Agreement (IAA) scores. Each target correction set can be divided into CONLL14 and BEA19 corrections. We evaluate the CONLL14 and BEA19 target corrections separately.

Correction	M2 Score	(Precision : Recall)
A1	46.9	44.6 : 59.1
A2	53.0	51.7 : 59.5
A3*	98.6	98.7 : 98.3
A4	55.3	54.9 : 57.0
A5	52.8	51.3 : 59.7
A6	56.4	55.8 : 58.8
A7*	98.6	98.7 : 98.5
A8	53.5	53.8 : 52.6
A9	55.7	55.6 : 56.0
A10	52.8	51.3 : 59.4
c1	50.9	49.0 : 60.4
c2	52.3	50.5 : 61.0
c3	53.7	52.1 : 60.8

Table 1: The M2 precision and recall quality score for all Bryant and Ng (2015) target correction sets for the official CONLL14 competition task.

Bryant and Ng (2015) released 10 additional target corrections for the CONLL14 test dataset. We compare the quality scores of our 3 corrections

with theirs using the official CONLL14 competition - M2 Scorer (Ng et al., 2014) metric. Table 1 shows the M2 scores for all target correction sets - Bryant and Ng (2015) corrections $A1 - A10$, and our corrections $c1 - c3$. Corrections $A3$ and $A7$ obtain near-perfect quality scores, since they were generated by the 2 editors who created the official CONLL14 competition target references (Bryant and Ng, 2015). Ignoring the 2 outliers, we observe similar quality scores for our corrections. This indicates that our 3 CONLL14 Target corrections are of similar high quality. Unfortunately, there are no public correction references available for the BEA19 Test set (this work being the first to present 3 target references), making it hard to compare the quality scores directly.

To overcome this issue, we calculate the quality scores for the 3 target correction sets and the GEC-Tool first-pass outputs on the official BEA19 and CONLL14 competitions and compare trends between the correction sets. We use the BEA19 competition website scorer⁴ to evaluate the performance of BEA19 target corrections. Table 2 shows the quality scores for the GEC-Tool and GEC-PD Tool output and the final editor target corrections (EC).

Similar trends are observed between the CONLL14 and BEA19 target correction sets. We observe a significant increase in Recall scores for the EC compared to the first-pass GEC Tool output. This indicates the final EC target contains additional post-edit corrections missed by the GEC Tool. The reduction in the precision score for EC is consistent with the 10 CONLL14 target corrections released by Bryant and Ng (2015) since post-editing often leads to subjective paraphrasing and rewrite edits, which may not be present in the official competition target reference. The final EC obtained better Recall scores compared to the State-of-the-Art (SOA) GEC Tool - GRECO (as of writing this paper) (Qorib and Ng, 2023) for both datasets. Observing similar quality score trends for the GEC Tool predictions and our target EC across both CONLL14 and BEA19 Test competition, and better Recall than the SOA GRECO tool, we can infer that the 3 target corrections collected by us in this work are of high quality.

We also use the GEC Inter Annotator Agreement (IAA) framework proposed by Bryant and

Ng (2015) to compare the target correction sets for both datasets with themselves to ensure better consistency and quality. The IAA framework states that the $F_{0.5}$ multi-reference score, used to evaluate a GEC Tool-vs-human corrections, can similarly evaluate human-vs-human corrections. When comparing multiple annotator corrections, a single correction set can be compared using the rest as references to get quality scores. The final IAA score is calculated as the average of all correction set scores. We use the ERRANT tool (Bryant et al., 2017) to perform the IAA evaluation. We evaluate 3 target correction sets:

$A = \{A1 - A10\}$ The 10 target corrections for CONLL14 by Bryant and Ng (2015).

$C = \{c1, c2, c3\}$ The 3 CONLL14 target corrections collected by us.

$B = \{b1, b2, b3\}$ The 3 BEA19 target corrections collected by us.

To compare IAA scores, we conduct a 1-vs-2 target correction set evaluation. For each correction in A , we randomly select 2 corrections from the remaining 9 as the reference. Scores for each correction in B and C are calculated using the remaining 2 corrections as target references. Table 4 shows the average IAA scores for A, B, C correction sets. We observe better Avg-IAA scores for the C and B correction sets collected by us in this work, compared to A .

To ensure we choose strong GEC Tools (Section 2.1) to obtain first-pass output predictions, we compare the quality of the GEC Tool output and the subsequent human EC. We consider the Source Sentence EC (collected by us) as the target reference for the BEA19 and CONLL14 Test sets. The $F_{0.5}$ quality scores obtained in Table 3 show similar performance between the GEC-Tool and GEC-PD Tool prediction output and the subsequent EC because of the variation in Precision and Recall scores. This indicates that GEC-Tool and GEC-PD are strong first-pass GEC Tools.

3.5 Impact of GEC Tools

Comparing the time-to-correct for the source sentence versus the GEC Tool output post-editing, we can quantify the impact of using GEC Tools in Text Editing.

Quality scores presented in Table 2 show that the GEC Tool output EC has better values compared to

⁴BEA19 GEC competition website - <https://codalab.lisn.upsaclay.fr/competitions/4057>

Candidate Set	BEA19 Test (P : R : $F_{0.5}$)	CONLL14 Test (P : R : $F_{0.5}$)
Source Sentence	-	-
Source Sentence EC	45.30 : 66.08 : 48.34	49.05 : 60.45 : 50.97
GECToR Output	66.81 : 58.42 : 64.94	63.97 : 45.94 : 59.31
GECToR Output EC	48.24 : 71.38 : 51.59	50.50 : 61.09 : 52.31
GEC-PD Output	66.20 : 61.48 : 65.20	64.06 : 44.92 : 59.03
GEC-PD Output EC	47.33 : 70.54 : 50.66	52.17 : 60.86 : 53.71
GRECO Model Output	86.45 : 63.13 : 80.50	79.36 : 48.69 : 70.48

Table 2: Quality Scores of the 2 GEC Tools output prediction, target Editor Corrections (EC) and the State-of-the-Art GEC Tool - GRECO (Qorib and Ng, 2023) on the official BEA19 and CONLL14 competition.

Candidate Set	BEA19 Test (P : R : $F_{0.5}$)	CONLL14 Test (P : R : $F_{0.5}$)
GECToR Output	52.59 : 28.59 : 45.03	57.74 : 25.10 : 45.82
GECToR Output EC	45.47 : 47.91 : 45.94	44.31 : 43.53 : 44.15
GEC-PD Output	49.88 : 26.37 : 42.33	56.49 : 23.13 : 43.85
GEC-PD Output EC	45.90 : 48.31 : 46.36	46.14 : 42.64 : 45.39

Table 3: Quality Scores of the 2 GEC Tools output predictions and their final target Editor Corrections (EC) using the BEA19 and CONLL14 - Source Sentence EC as target reference.

Human Annotation Set	Reference Set and Size	IAA Score - $F_{0.5}$
A1	$ \{RAND(2) \in \{A - A1\}\} = 2$	36.21
A2	$ \{RAND(2) \in \{A - A2\}\} = 2$	45.48
A3	$ \{RAND(2) \in \{A - A3\}\} = 2$	46.72
A4	$ \{RAND(2) \in \{A - A4\}\} = 2$	40.54
A5	$ \{RAND(2) \in \{A - A5\}\} = 2$	46.01
A6	$ \{RAND(2) \in \{A - A6\}\} = 2$	50.85
A7	$ \{RAND(2) \in \{A - A7\}\} = 2$	42.72
A8	$ \{RAND(2) \in \{A - A8\}\} = 2$	49.46
A9	$ \{RAND(2) \in \{A - A9\}\} = 2$	52.0
A10	$ \{RAND(2) \in \{A - A10\}\} = 2$	48.57
Avg-IAA $\{A\}$	$\{A\}, 2$	45.85
c1	$ \{C - c1\} = 2$	54.11
c2	$ \{C - c2\} = 2$	57.36
c3	$ \{C - c3\} = 2$	59.14
Avg-IAA $\{C\}$	$\{C\}, 2$	56.87
b1	$ \{B - b1\} = 2$	57.94
b2	$ \{B - b2\} = 2$	59.39
b3	$ \{B - b3\} = 2$	59.81
Avg-IAA $\{B\}$	$\{B\}, 2$	59.05

Table 4: Inter Annotator Agreement (IAA) scores for the different A, B, C annotation sets using the ERRANT $F_{0.5}$ metric. RAND(n) represents a random selection of "n" items from the respective set.

the Source Sentence EC. In Table 5, we compare the time taken (in seconds) by a human editor to correct the source sentences with and without first-pass editing by a GEC tool. We observe that GEC Tools help in reducing the post-editing time by

roughly 4 seconds per sentence. Combined insights from these results indicate that incorporating GEC Tools in the text-editing workflow reduces editing time and generates better final target corrections. Thus, GEC Tools can help improve editor efficiency

Sentence Source	Average Time per Sentence	Average Time per Word
Source Sentence	31.16	1.91
GECtoR Output	26.82	1.57
GEC-PD Output	27.46	1.67

Table 5: The average time to correct (**in seconds**) for a sentence and word; correcting the source and after first-pass GEC Tool editing.

and overall productivity.

4 Methodology

We design statistical and neural network regression models for our post-editing effort in time (PEET) scorer. The scorer is trained to estimate the time-to-correct value for a source sentence given the target correction, using the number and type of edits and sentence property - Sentence Length, Correct/Incorrect.

The dataset that we collected contains 3 iterations for all 3 variations of the source - source (SRC), GEC Tool Model Output (MO) and post-edited target correction (TRG). Different training features in terms of edits and sentence structure can be selected and extracted from - SRC, MO and TRG triple (Appendix D).

Statistical PEET models performed as well as Neural models while allowing greater interpretability of training features (Appendix A). Also, models using features selected from $[MO, TRG]$ sentences performed better than models trained on fine-grained features from $[SRC, MO, TRG]$ sentences (Appendix E). Hence, we only discuss the features and results of the Statistical PEET Model trained using the $[MO, TRG]$ sentences here, referring to MO as the source.

4.1 ERRANT Edit Feature Extraction

We use ERRANT (Bryant et al., 2017) to align and extract edit features between the source and target corrections (Appendix B). Apart from the edit category - Removal(R), Missing(M) and Unnecessary(U), the feature also includes the edit type. Figure 1 lists the different edit categories and their syntactic type generated by ERRANT.

We use the number and type of edits as features for our statistical models. Similar to the edit type

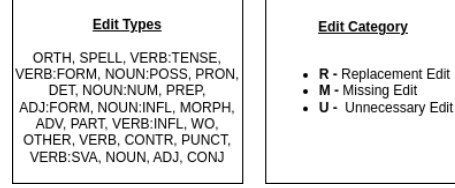


Figure 1: ERRANT edit category and types.

hierarchy used by Yuan et al. (2021), considering category, type and their combination can provide 4, 25 or 55 edit features. For instance, if we only consider the 3 edit categories, then our 4 edit features are Replacement(R), Missing(M), Unnecessary(U) and Correct/Incorrect (binary feature). Using the 24 edit types (Figure 1) and Correct/Incorrect gives us 25 edit features. Similarly, combining edit categories with their possible types, we get 55 edit features (see Table 14 in Appendix G). We train separate models for all three edit levels (4, 25, 55).

4.2 PEET Scorer Models

We design Linear Regression (LR) and Support Vector Regression (SVR) models, for our PEET Scorer, using the ERRANT Edit count and different edit type levels (4, 25, 55), number of edited words, source and target sentence length as features. We also experimented with Neural Regression models, but they didn't perform better than statistical models (Results in Appendix A). We only discuss the results of the statistical PEET models here. The details of each model and the hyperparameters are presented in Appendix C.

The PEET estimation task has a continuous range of prediction values - time (in seconds). We report the mean absolute error (MAE) and Pearson correlation (r) between the predicted time and the target time. We note that MAE does not take into account the sign of the error, while correlation does (Graham, 2015; Tezcan et al., 2019), which is why we report correlation and use it to compare model performance.

5 Experiment Results

5.1 Performance of the PEET Scorer

The results for the Linear Regression (LR) and SVR PEET Scorer, with count of different edit feature levels (4,25,55), sentence word length and number of word edits as features (Section 4.1), are presented in Table 6.

The statistical models relying on edit type information (25,55 labels) performed better than using

Statistical Model	Edit Feature Level	r	MAE
Linear Regression	4	0.559	18.92
	25	0.565	18.74
	55	0.563	18.75
SVR Linear	4	0.558	16.40
	25	0.564	16.19
	55	0.565	16.15

Table 6: Average PEET estimation performance for the Statistical Models over 50 runs (different train-test data seed). The results are presented as the Pearson Correlation (r), Mean Absolute Error (MAE) loss.

minimal substitution, deletion and insertion edit category labels (Figure 1). This indicates that the type of edit has an impact on post-editing effort. We obtain a correlation of $r = 0.565$ from the best models (LR-25 edit features).

5.2 Impact of Error Types on Post-Edit Effort

We follow the work by Ye et al. (2021), using regression coefficients of a Linear Regression (LR) model to estimate the PEET impact of different edit features. To make the coefficients interpretable, we center and standardize all edit-features by subtracting the mean and dividing by the standard deviation (except the binary/categorical edit feature - Correct/Incorrect) (Schielzeth, 2010).

Model Features	Regression Coefficient	Model Features	Regression Coefficient	Model Features	Regression Coefficient
OTHER	10.15	ORTH	2.34	ADJ	0.97
PUNCT	4.55	CONJ	2.03	CONTR	0.78
PREP	4.03	MORPH	1.89	VERB:INFL	0.63
VERB	3.37	SPELL	1.87	PART	0.47
Sentence Correct	-3.31	ADV	1.79	ADJ:FORM	0.39
NOUN	3.23	VERB:FORM	1.66	NOUN:INFL	-0.30
DET	3.08	WO	1.63	NOUN:POSS	0.25
NOUN:NUM	2.52	VERB:SVA	1.16	-	-
VERB:TENSE	2.35	PRON	1.10	-	-

Table 7: The standardized regression coefficients of the LR model trained on the medium (25) edit features to measure the impact of each feature on PEET estimation.

The edit category *OTHER*, which corresponds to paraphrasing or rewriting text, and modifying punctuation has the highest impact on post editing time. Deciding whether a particular sentence is incorrect also contributes significantly to the post-editing effort. The coefficients to study the impact of the 25 edit features are shown in Table 7. Coefficients for the other edit granularities (4 and 55 labels) and all PEET sentence features are provided in Appendix G.

5.3 PEET Scorer for GEC Quality Estimation

Since an efficient GEC Tool would reduce post-editing (PE) time, PE followed by PEET estimation can quantify the usability of a GEC Tool (Specia, 2011). To study the correlation between cognitive, temporal and technical PE effort, we compare the PEET scorer rankings with human judgment rankings (HJR) (Section 2.3) and Word Error Rate (Technical Effort) of GEC Tools. We evaluate the PEET-Linear Regression (25 Edit Features) Scorer (Section 4.1) estimated ranking for 33 GEC Tools in 3 GEC HJR (Appendix H).

- *Grundkiewicz-C14(EW)* - ranking of 12 GEC Tools that participated in the official CONLL-14 - GEC Task (Ng et al., 2014) by Grundkiewicz et al. (2015).
- *SEEDA-C14-All(TS)* - ranking of 15 newer and stronger GEC Tools on the CONLL-14 test dataset by Kobayashi et al. (2024). *SEEDA-C14-NO(TS)* denotes the subset of 12 GEC tools without the 3 outliers.
- *Napoles-FCE* and *Napoles-Wiki* - ranking of 6 Seq2Seq GEC Tools on the FCE (Yannakoudakis et al., 2011) and WikiEd (Grundkiewicz and Junczys-Dowmunt, 2014) datasets by Napoles et al. (2019).

Human Judgment Ranking	PEET Metric		WER	
	ρ	r	ρ	r
Grundkiewicz - C14 (EW)	0.48	0.26	0.28	0.18
SEEDA - C14 - All (TS)	0.18	0.63	0.18	0.65
SEEDA - C14 - NO (TS)	-0.1	-0.27	-0.1	-0.33
Napoles - FCE	-0.96	-0.94	-0.96	-0.88
Napoles - Wiki	-0.71	-0.63	-0.93	-0.88

Table 8: The correlation of our PEET model ranking with human-judgment rankings (HJR). We also provide the correlation of the HJR with the Word Edit Rate (WER) metric. Spearman (ρ) and Pearson (r) correlation scores are used for comparison. A high negative correlation indicates lower time-to-correct and WER score corresponding to a higher human judgment ranking.

The *Grundkiewicz-C14* and *SEEDA-C14* human ranking calculation was conducted using the Expected Wins (EW) (Bojar et al., 2013) and TrueSkill (TS) (Herbrich et al., 2007) method, which tracks relative ranking based on a set-wise comparison of a subset of all GEC Tool corrections. The EW and TS rankings were selected for the final *Grundkiewicz-C14* and *SEEDA-C14* rankings,

respectively. The *Napoles - FCE* and *Napoles - Wiki* human ranking addressed the issue of partial comparison and relative ranking for GEC Tools by using the partial ranking with scalars (PRWS) method (Sakaguchi and Van Durme, 2018), collecting a quality score (0-100) for each sentence to infer the final rankings.

Table 8 shows the Pearson (r) and Spearman (ρ) correlation scores of the HJRs with the PEET model ranking and the Word Error Rate (WER) (Snover et al., 2006) (number of edits required to correct a GEC Tool prediction). The WER and PEET are calculated using untargeted references, which contributes to the lower alignment with perceived cognitive effort judgment.

We observe a good alignment (high negative correlation) between the PEET ranking and the *Napoles* HJR and a poor alignment (positive correlation) with the other HJRs. The PEET ranking shows better alignment to HJRs that align with WER scores (Technical PE effort - Section 2.3). We also observe that human quality rankings collected using PRWS align better with true human effort (WER) than those collected using TS or EW.

These results suggest that our PEET Scorer can estimate GEC Tool usability when output quality depends on further Post-Editing Effort (WER and type of edits) required to correct the tool output. Hence, performing PE to obtain the closest correction (lower WER) can improve GEC temporal effort estimation.

6 Conclusion and Future Work

Since we present the first study and dataset of Post-Editing Effort (PEET) in Time for GEC, our goal is to provide a baseline for future work in this area. Using our dataset, we quantified the editor efficiency and productivity when using GEC Tools for Text Editing. We extract various automated sentence properties and edit type features from the sentence correction pairs to train the PEET Scorer. Recently, there has been some work in the area of Grammar Error Explanation to define descriptive error types (Fei et al., 2023; Ye et al., 2025) and use LLMs for error explanation (Song et al., 2023; Li et al., 2025). As future work, the descriptive edits can be used as possible features for the PEET model. Finally, we observe that our PEET model works well for GEC Tool evaluation when the output quality is dependent on the Technical PE Effort (amount of edits). Studying actual cognitive effort

for GEC post-editing and how it compares with technical and temporal effort is another interesting direction for future work.

Limitations

One of the main limitations of Post-Editing (PE) Effort estimation is incorporating human annotation to evaluate GEC Tool performance, which can be expensive. However, PE allows us to quantify the true performance from a human-in-the-loop perspective. Currently, our work is limited to automated edit-type features generated by the ERRANT toolkit (Bryant et al., 2017). Evaluating our PEET Scorer as a GEC quality estimation tool shows that it is effective when the correction quality is dependent on the technical post-editing effort. However, similar to work in Machine Translation, it is inconsistent with quality estimation based on perceived PE efforts. Finally, we acknowledge that our work is limited to only the English language. Future work on post-editing GEC for other languages can show the impact of language type on PEET for GEC.

Acknowledgments

We thank staff and colleagues at Scribendi Inc., Chatham, Ontario (www.scribendi.com) for the grant, input and feedback during the research and manuscript writing phases for this project. Resources used in this work were provided by the Province of Ontario, the Government of Canada through CIFAR, companies sponsoring the Vector Institute (<https://vectorinstitute.ai/partnerships/current-partners/>), the Natural Sciences and Engineering Council of Canada and a grant from IITP & MSIT of Korea (No. RS-2024-00457882, AI Research Hub Project).

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). 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 4260–4270.
- Srinivas Bangalore, Bergljot Behrens, Michael Carl, Maheshwar Gankhot, Arndt Heilmann, Jean Nitzke, Moritz Schaeffer, and Annegret Sturm. 2015. [The role of syntactic variation in translation and post-editing](#). *Translation Spaces*, 4(1):119–144.

- Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and 1 others. 2009. [Statistical approaches to computer-assisted translation](#). *Computational Linguistics*, 35(1):3–28.
- Ondrej 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 *WMT@ACL*.
- Tiberiu Boroş, Stefan Daniel Dumitrescu, Adrian Zafiu, Verginica Barbu Mititelu, and Ionut Paul Văduva. 2014. [Racai gec—a hybrid approach to grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 43–48.
- 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.
- 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.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. [Grammatical error correction: A survey of the state of the art](#). *Computational Linguistics*, 49(3):643–701.
- CJ Bryant, Mariano Felice, and Edward Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). Association for Computational Linguistics.
- Michael Carl, Barbara Dragsted, Jakob Elming, Daniel Hardt, and Arnt Lykke Jakobsen. 2011. The process of post-editing: A pilot study. *Copenhagen Studies in Language*, 41(1):131–142.
- Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeol Yoon. 2019. [A neural grammatical error correction system built on better pre-training and sequential transfer learning](#). *arXiv preprint arXiv:1907.01256*.
- Shamil Chollampatt and Hwee Tou Ng. 2018. [A multi-layer convolutional encoder-decoder neural network for grammatical error correction](#). In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Napoles Courtney, Sakaguchi Keisuke, Post Matt, R Tetreault Joel, and 1 others. 2016. [Gleu without tuning](#). *arXiv*.
- Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. [Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction](#). *arXiv preprint arXiv:2303.14342*.
- Joke Daems, Sonia Vandepitte, Robert J Hartsuiker, and Lieve Macken. 2017. [Identifying the machine translation error types with the greatest impact on post-editing effort](#). *Frontiers in psychology*, 8:1282.
- Joke Daems, Sonia Vandepitte, Robert Hartsuiker, and Lieve Macken. 2015. The impact of machine translation error types on post-editing effort indicators. In *Proceedings of the 4th Workshop on Post-editing Technology and Practice*.
- 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.
- Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. [Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1816–1829.
- Christopher Davis, Andrew Caines, Øistein Andersen, Shiva Taslimipour, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of english learner text](#). *arXiv preprint arXiv:2401.07702*.
- 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.
- Tao Fang, Xuebo Liu, Derek F Wong, Runzhe Zhan, Liang Ding, Lidia S Chao, Dacheng Tao, and Min Zhang. 2023a. [Transgec: Improving grammatical error correction with translationese](#). In *Findings of the association for computational linguistics: ACL 2023*, pages 3614–3633.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023b. [Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation](#). *arXiv preprint arXiv:2304.01746*.
- Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. [Enhancing grammatical error correction systems with explanations](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

- Mariano Felice, Zheng Yuan, Øistein E Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. [Grammatical error correction using hybrid systems and type filtering](#). In *Proceedings of the eighteenth conference on computational natural language learning: shared task*, pages 15–24.
- Peiyuan Gong, Xuebo Liu, He-Yan 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.
- 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.
- Yvette Graham. 2015. [Improving evaluation of machine translation quality estimation](#). 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 1804–1813.
- Spence Green, Jeffrey Heer, and Christopher D Manning. 2013. [The efficacy of human post-editing for language translation](#). In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 439–448.
- Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. [The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction](#). In *Advances in Natural Language Processing: 9th International Conference on NLP, PolTAL 2014, Warsaw, Poland, September 17-19, 2014. Proceedings 9*, pages 478–490. Springer.
- 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.
- Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.
- Ana Guerberof. 2009. Productivity and quality in mt post-editing. In *Beyond Translation Memories: New Tools for Translators Workshop*.
- Anubhav Gupta. 2014. [Grammatical error detection using tagger disagreement](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 49–52, Baltimore, Maryland. Association for Computational Linguistics.
- Ralf Herbrich, Tom Minka, and Thore Graepel. 2007. [Trueskill™: A bayesian skill rating system](#). In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*. The MIT Press.
- S. David Hernandez and Hiram Calvo. 2014. [CoNLL 2014 shared task: Grammatical error correction with a syntactic n-gram language model from a big corpora](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 53–59, Baltimore, Maryland. Association for Computational Linguistics.
- Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.
- Kristian Tangsgaard Hvelplund. 2014. Eye tracking and the translation process: Reflections on the analysis and interpretation of eye-tracking data.
- 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.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. [The amu system in the conll-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33.
- Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. [Phrase-based machine translation is state-of-the-art for automatic grammatical error correction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556.
- Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. [Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- 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.
- Nikita Kitaev and Dan Klein. 2018. [Constituency parsing with a self-attentive encoder](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

- Richard Kittredge. 2002. [Krings, hans p. \(2001\): Repairing texts: Empirical investigations of machine translation post-editing processes \(geoffrey s. koby, ed.\), the kent state university press, kent, ohio london, 558 p. Meta, 47\(3\):435–436.](#)
- Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. [An empirical study of incorporating pseudo data into grammatical error correction.](#) 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 1236–1242.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M Rush. 2018. [Opennmt: Neural machine translation toolkit.](#) In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 177–184.
- Masamune Kobayashi, Masato Mita, and Mamoru Komachi. 2024. [Revisiting meta-evaluation for grammatical error correction.](#) *Transactions of the Association for Computational Linguistics*, 12:837–855.
- Maarit Koponen. 2012. Comparing human perceptions of post-editing effort with post-editing operations. In *Proceedings of the seventh workshop on statistical machine translation*, pages 181–190.
- Maarit Koponen. 2016. [Is machine translation post-editing worth the effort? a survey of research into post-editing and effort.](#) *The Journal of Specialised Translation*, (25):131–148.
- Maarit Koponen, Wilker Aziz, Luciana Ramos, Lucia Specia, Jussi Rautio, Lauri Carlson, Inari Listienmaa, Seppo Nyrkkö, Gorka Labaka, Arantza Díaz De Ilaraza, and 1 others. 2012. Post-editing time as a measure of cognitive effort. In *AMTA 2012 Workshop on Post-editing Technology and Practice (WPTP)*.
- Oleksandr Korniienko. 2024. Enhancing grammatical correctness: The efficacy of large language models in error correction task.
- Anoop Kunchukuttan, Sriram Chaudhury, and Pushpak Bhattacharyya. 2014. [Tuning a grammar correction system for increased precision.](#) In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 60–64.
- Vivian Lai, Alison Smith-Renner, Ke Zhang, Ruijia Cheng, Wenjuan Zhang, Joel Tetreault, and Alejandro Jaimes-Larrarte. 2022. [An exploration of post-editing effectiveness in text summarization.](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 475–493, Seattle, United States. Association for Computational Linguistics.
- Kyusong Lee and Gary Geunbae Lee. 2014. [Postech grammatical error correction system in the conll-2014 shared task.](#) In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 65–73.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.
- Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. 2024. [Dissecting human and llm preferences.](#) *arXiv preprint arXiv:2402.11296*.
- Wei Li, Wen Luo, Guangyue Peng, and Houfeng Wang. 2025. [Explanation based in-context demonstrations retrieval for multilingual grammatical error correction.](#) *arXiv preprint arXiv:2502.08507*.
- Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F Wong, Yang Gao, He-Yan Huang, and Min Zhang. 2023. [Templategec: Improving grammatical error correction with detection template.](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6878–6892.
- 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.](#) *arXiv preprint arXiv:1907.11692*.
- 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.
- Maria Nadejde and Joel Tetreault. 2020. [Personalizing grammatical error correction: Adaptation to proficiency level and ll.](#) *arXiv preprint arXiv:2006.02964*.
- 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, and Joel Tetreault. 2017. [Jfleg: A fluency corpus and benchmark for grammatical error correction.](#) *arXiv preprint arXiv:1702.04066*.
- 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.

- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashnyi. 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.
- Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashnyi, Artem Chernodub, Oleksandr Kornienko, 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.
- Robert Östling, Katarina Gillholm, Murathan Kurfalı, Marie Mattson, and Mats Wirén. 2024. [Evaluation of really good grammatical error correction](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6582–6593, Torino, Italia. ELRA and ICCL.
- Sharon O’Brien. 2005. [Methodologies for measuring the correlations between post-editing effort and machine translatability](#). *Machine translation*, 19:37–58.
- Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020a. [Adapterfusion: Non-destructive task composition for transfer learning](#). *arXiv preprint arXiv:2005.00247*.
- Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020b. [Adapterhub: A framework for adapting transformers](#). *arXiv preprint arXiv:2007.07779*.
- Mirko Plitt and François Masselot. 2010. [A productivity test of statistical machine translation post-editing in a typical localisation context](#). *Prague Bull. Math. Linguistics*, 93:7–16.
- Maja Popović, Arle Lommel, Aljoscha Burchardt, Eleftherios Avramidis, and Hans Uszkoreit. 2014. Relations between different types of post-editing operations, cognitive effort and temporal effort. In *Proceedings of the 17th annual conference of the european association for machine translation*, pages 191–198.
- Muhammad Reza Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. [Frustratingly easy system combination for grammatical error correction](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974.
- Muhammad Reza Qorib and Hwee Tou Ng. 2023. [System combination via quality estimation for grammatical error correction](#). *arXiv preprint arXiv:2310.14947*.
- Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. [Coedit: Text editing by task-specific instruction tuning](#). *arXiv preprint arXiv:2305.09857*.
- 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.
- Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. [The illinois-columbia system in the conll-2014 shared task](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 34–42.
- Alla Rozovskaya and Dan Roth. 2010. [Annotating ESL errors: Challenges and rewards](#). In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California. Association for Computational Linguistics.
- Alla Rozovskaya and Dan Roth. 2021. [How good \(really\) are grammatical error correction systems?](#) In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2686–2698.
- Keisuke Sakaguchi and Benjamin Van Durme. 2018. [Efficient online scalar annotation with bounded support](#). *arXiv preprint arXiv:1806.01170*.
- Holger Schielzeth. 2010. [Simple means to improve the interpretability of regression coefficients](#). *Methods in Ecology and Evolution*, 1(2):103–113.
- Dorothy Senez. 1998. Post-editing service for machine translation users at the european commission. In *Proceedings of Translating and the Computer 20*.
- Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. [The University of Edinburgh’s neural MT systems for WMT17](#). In *Proceedings of the Second Conference on Machine Translation*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.
- Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. [A study of translation edit rate with targeted human annotation](#). In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.
- Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer. 2023. [Gee! grammar error explanation with large language models](#). *arXiv preprint arXiv:2311.09517*.

- Lucia Specia. 2011. [Exploiting objective annotations for minimising translation post-editing effort](#). In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*, Leuven, Belgium. European Association for Machine Translation.
- Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *Proceedings of Machine Translation Summit XIII: Papers*.
- Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*.
- Somayajulu Sripada, Ehud Reiter, and Lezan Hawizy. 2005. [Evaluation of an NLG system using post-edit data: Lessons learnt](#). In *Proceedings of the Tenth European Workshop on Natural Language Generation (ENLG-05)*, Aberdeen, Scotland. Association for Computational Linguistics.
- Yujin Takahashi, Masahiro Kaneko, Masato Mita, and Mamoru Komachi. 2022. [ProQE: Proficiency-wise quality estimation dataset for grammatical error correction](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5994–6000, Marseille, France. European Language Resources Association.
- Maksym Tarnavskyy, 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.
- Midori Tatsumi. 2009. Correlation between automatic evaluation metric scores, post-editing speed, and some other factors. In *Proceedings of Machine Translation Summit XII: Posters*.
- Midori Tatsumi and Johann Roturier. 2010. Source text characteristics and technical and temporal post-editing effort: what is their relationship. In *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 43–52.
- Irina P Temnikova. 2010. Cognitive evaluation approach for a controlled language post-editing experiment. In *LREC*.
- Joel Tetreault and Martin Chodorow. 2008. [Native judgments of non-native usage: Experiments in preposition error detection](#). In *Coling 2008: Proceedings of the workshop on human judgements in computational linguistics*, pages 24–32.
- Joel Tetreault, Martin Chodorow, and Nitin Madnani. 2014. [Bucking the trend: improved evaluation and annotation practices for esl error detection systems](#). *Language Resources and Evaluation*, 48:5–31.
- Arda Tezcan, Veronique Hoste, and Lieve Macken. 2016. [Detecting grammatical errors in machine translation output using dependency parsing and treebank querying](#). In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 203–217.
- Arda Tezcan, Véronique Hoste, and Lieve Macken. 2019. [Estimating post-editing time using a gold-standard set of machine translation errors](#). *Computer Speech & Language*, 55:120–144.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *arXiv preprint arXiv:2307.09288*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). NIPS’17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Lucas Nunes Vieira. 2014. [Indices of cognitive effort in machine translation post-editing](#). *Machine translation*, 28(3):187–216.
- Lucas Nunes Vieira. 2017. [Cognitive effort and different task foci in post-editing of machine translation: A think-aloud study](#). *Across Languages and Cultures*, 18(1):79–105.
- Peilu Wang, Zhongye Jia, and Hai Zhao. 2014a. [Grammatical error detection and correction using a single maximum entropy model](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 74–82.
- Yiming Wang, Longyue Wang, Xiaodong Zeng, Derek F Wong, Lidia S Chao, and Yi Lu. 2014b. [Factored statistical machine translation for grammatical error correction](#). In *Proceedings of the eighteenth conference on computational natural language learning: Shared task*, pages 83–90.
- Jian-Cheng Wu, Tzu-Hsi Yen, Jim Chang, Guan-Cheng Huang, Hsiang-Ling Hsu, Yu-wei Chang, and Jason S Chang. 2014. [Nthu at the conll-2014 shared task](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 91–95.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.
- Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. [Lm-critic: Language models for unsupervised grammatical error correction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7752–7763.

Jingheng Ye, Shang Qin, Yinghui Li, Hai-Tao Zheng, Shen Wang, and Qingsong Wen. 2025. [Corrections meet explanations: A unified framework for explainable grammatical error correction](#). *arXiv preprint arXiv:2502.15261*.

Na Ye, Ling Jiang, Dandan Ma, Yingxin Zhang, Sanyuan Zhao, and Dongfeng Cai. 2021. [Predicting post-editing effort for english-chinese neural machine translation](#). In *2021 International Conference on Asian Language Processing (IALP)*, pages 154–158. IEEE.

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiura, 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.

Zheng Yuan, Shiva Taslimipour, Christopher Davis, and Christopher Bryant. 2021. [Multi-class grammatical error detection for correction: A tale of two systems](#). In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 8722–8736.

Anna Zaretskaya, Mihaela Vela, Gloria Corpas Pastor, and Miriam Seghiri. 2016. Measuring post-editing time and effort for different types of machine translation errors.

Longkai Zhang and Houfeng Wang. 2014. [A unified framework for grammar error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 96–102.

Yue Zhang, Leyang Cui, Deng Cai, Xinting Huang, Tao Fang, and Wei Bi. 2023. [Multi-task instruction tuning of llama for specific scenarios: A preliminary study on writing assistance](#). *arXiv preprint arXiv:2305.13225*.

A Neural Regression Models for PEET Estimation

Since semantics and syntax structure have been shown to impact PE effort (Tezcan et al., 2016; Bangalore et al., 2015), we also trained neural-LM PEET Scorer models using flattened constituency parse trees (Kitaev and Klein, 2018) and part-of-speech syntax structure features for the source and target corrections, generated using the spaCy library (Honnibal and Montani, 2017).

Model Features	BERT-L		RoBERTa-L	
	r	MAE	r	MAE
Sentence Edit	0.552	17.73	0.56	17.97
Syntactic Variation	0.528	19.35	0.564	18.05
#EW + Syntactic Variation	0.564	17.16	0.561	16.88
#EW + Syntax Structure	0.565	18.57	0.565	18.74

Table 10: Performance of Neural PEET models using different sequence model features over 5 runs. The results are shown as Pearson Correlation (r) and Mean Absolute Error (MAE) loss.

Pretrained LMs can also capture syntax structure internally (Dai et al., 2021), so we also train neural-LM models using only source-target sentence embeddings as features to estimate PEET. Since the statistical models work as well as Neural models, while being faster and more interpretable, we consider them for the PEET Scorer in the main paper. We describe the features (Table 9) and results of the Neural PEET (Table 10) model here.

B GEC Evaluation File Example and Format

The evaluation of a GEC Tool requires a Source (S), Target (T) and Model Output (MO) sentence. Table 11 gives an example of such a triple. GEC evaluation generates M2 file for a pair of sentences (e.g., S and T), which lists the edits that can convert sentence S into sentence T and the positions of those edits. The evaluation process generates two M2 files : (Source - Target) and (Source - Model Output). The M2 edits are compared to evaluate the Model Output quality.

• Source-Target M2 File:

S Surrounded by such concerns , it is very likely that we are distracted to worry about these problems .

A 13 14|||R:OTHER|||and|||REQUIRED||| -NONE-|||0
A 11 12|||R:VERB:TENSE|||will be|||REQUIRED|||
-NONE-|||1

A 12 12|||M:ADV|||too|||REQUIRED||| -NONE-|||1

• Source-Model Output M2 File:

S Surrounded by such concerns , it is very likely that we are distracted to worry about these problems .

A 13 14|||R:PART|||from|||REQUIRED||| -NONE-|||0
A 14 15|||R:VERB:FORM|||worrying||| REQUIRED|||
-NONE-|||0

The M2 file format was part of the GEC-M2 Scorer evaluation tool proposed by Dahlmeier and Ng (2012). The tool generates an alignment and

Model Type	Input Format
Sentence Edit	[MO] <mo-sentence> [TRG] <trg- sentence>
Syntactic Variation	<mo-constituency-parse> [TO] <trg-constituency-parse>
#EW + Syntactic Variation	#EW - <mo-constituency-parse> [TO] <trg-constituency-parse>
#EW + Syntax Structure	#EW - <trg-part-of-speech-tag>

Table 9: The training data format for the BERT and RoBERTa LM. The example considers a sentence pair - <mo-sentence> and <trg- sentence> where "mo" is the Model Output correction made by a GEC Tool and the "trg" is the post-edited target correction for "mo". The special tokens [MO], [TRG] and [TO] denote sentence breaks in the input. #EW denotes the number of edited words between mo and trg.

Source :	Surrounded by such concerns, it is very likely that we <u>are</u> distracted to worry about these problems.
Target :	Surrounded by such concerns, it is very likely that we <u>will be too</u> distracted to worry about these problems.
Model Output :	Surrounded by such concerns, it is very likely that we are distracted from <u>worrying</u> about these problems.

Table 11: *Source*, *Target* and example *Model Output* made by a GEC Tool.

detects atomic edits between a pair of sentences. Further improvement to the M2 tool was done by Bryant et al. (2017), resulting in the ERRANT evaluation tool. The ERRANT tool retained the overall M2 file format, utilizing syntactic and linguistic features to extract better-aligned and tagged edits between 2 sentences (as shown above).

C Predictive Model Parameters

We train different statistical and neural predictive models to estimate the post-editing temporal effort. We use this section to describe the predictive models as well as the training parameters for the regression task.

Linear Regression: We use the Linear Regression (LR) model provided by the Scikit-Learn library⁵. To keep the weights of the features from getting arbitrarily high, we used the RidgeLinear model that also adds an L2 Regularizer to the model. We trained the model with default training parameters and $\alpha = 1.0$.

⁵https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

Support Vector Regression: We also train Support Vector Regression (SVR) models from scikit-learn with the default training parameters and the "linear" kernel.

BERT, RoBERTa Neural Models: To train neural predictive models, we fine-tuned the BERT-Large (Devlin et al., 2019) and RoBERTa-Large (Liu et al., 2019) with a regression head. The models were trained using the Pfeiffer bottleneck adapters (Pfeiffer et al., 2020a) which allowed us to reduce the training time. We utilized the AdapterHub library⁶ for training the models with the default Pfeiffer adapter configuration (Pfeiffer et al., 2020b). Training was done for 50 epochs with a 10-epoch and .05 loss threshold early stopping. A learning rate of $1e - 04$ was used. To train the models for the regression task, we added a one-label regression head and used the mean-square-error loss (MSELoss), which is part of the Huggingface⁷ training pipeline.

D Different Sources for Training Feature Selection and Extraction

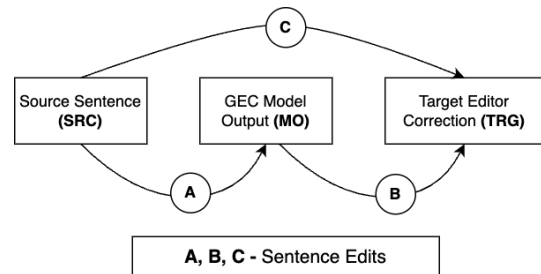


Figure 2: Sentence correction edits extracted using the ERRANT toolkit.

Our dataset has 3 iterations for each source sentence. We have the original sentence - source

⁶<https://adapterhub.ml/>

⁷<https://huggingface.co/>

(SRC), the first-pass correction by a GEC Tool - Model Output (MO) and the final targeted editor correction - target (TRG). Figure 2 shows the 3 iterations for the source sentence. Each arc represents a sentence transition pairing and can be used to extract intermediate edit features. To extract features, the following sentence pairings can be considered: [MO], [SRC - MO], [MO - TRG], [SRC - MO - TRG]. Post-editing features, from different levels, can be extracted from the $SRC - MO - TRG$ and $MO - TRG$ sentence pairings. Considering the source sentence as a feature can further separate target edits into ignored and incorrect edits.

- **SRC - MO - TRG:** We consider and extract the set of edits - A and C (Figure 2) for the model features. We further use these edits to create 2 categories - Incorrect and Ignored edits.

- Incorrect: $|A - C|$
- Ignored: $|C - A|$

- **MO - TRG:** We consider only edit set - B (Figure 2) as the input for the trained models.

We found that the performance of models trained on these 2 feature sources was comparable (Appendix E). This also indicates that the PEET Scorer can estimate time-to-correct from the post-editing correction stage - B. We only present and discuss the results of the model trained using the $MO - TRG$ sentence features in the main paper. Results for the $[SRC - MO - TRG]$ Scorer are presented in Appendix E.

E PEET Scorer using SRC, MO and TRG Sentence Features

Model Features	BERT-L		RoBERTa-L	
	r	MAE	r	MAE
Sentence Edit	0.513	19.10	0.54	17.82

Table 12: Neural PEET model performance over 5 runs using the source (SRC), GEC Tool Model Output (MO) and Target Correction (TRG) sentence features. The results are shown as Pearson Correlation (r) and Mean Absolute Error (MAE) loss.

Statistical Model	Edit Feature Level	r	MAE
Linear Regression	10	0.558	18.92
	106	0.557	18.89
SVR Linear	10	0.556	16.39
	106	0.561	16.21

Table 13: PEET Statistical Model performance over 50 runs (different train-test data seed) using Incorrect and Ignored separated Edit features (Appendix D) extracted from SRC, MO and TRG sentence triples. The results are presented as the Pearson Correlation (r), Mean Absolute Error (MAE) loss.

F GEC Post Editing Instructions and Survey Example

Welcome to the Sentence Checking and Correction Survey.

Task: This survey will ask you to evaluate and validate around 50 sentences. A sentence correction system corrected and generated these sentences. The survey will provide the original source sentence for comparison along with the correction in a text-box underneath. If required, please make further minimal edits in the text-box, while preserving the original sentence's meaning. You do not need to complete the survey all at once. You can continue the survey after a break or reload the URL to be presented with the next sentence you need to evaluate. This survey is anonymous and it will not collect any personal information.

Steps:

- For each presented sentence, if required, make corrections in the text box.
- Please proofread the sentences. We need to correct the sentences in the survey by making minimal corrections. Please avoid rewriting/rephrasing the sentence.
- Once satisfied, press the submit button.

Figure 3: Survey instructions for the editor to perform post editing, and obtain target corrections for our dataset.

① Fortunately, the hippo mother gave a new birth last month.

Fortunately, the hippo mother gave birth last month.②

③

- 1 - Original Source Sentence
- 2 - First-pass GEC-Model Correction
- 3 - Textbox for Editor final review on 2.

Figure 4: Example source sentence and its first-pass edit from the Survey. The editor can make further improvements in the text box. Submitting the final target correction.

G Feature Impact on Post-Editing Time using Regression Coefficients

We utilize the regression coefficients of a Ridge-Linear Regression model to quantitatively calculate the impact of different edit type features on the

Model Features	Regression Coefficient	Model Features	Regression Coefficient	Model Features	Regression Coefficient	Model Features	Regression Coefficient	Model Features	Regression Coefficient
R:OTHER	7.73	M:DET	2.03	M:VERB	1.49	U:VERB	1.07	M:ADJ	0.36
U:OTHER	4.53	M:OTHER	1.98	R:VERB:FORM	1.48	M:ADV	0.93	R:CONJ	0.30
Sentence Correct	-3.11	R:DET	1.94	U:PUNCT	1.36	U:ADJ	0.79	U:NOUN:POSS	0.29
R:PREP	2.85	M:PREP	1.93	U:ADV	1.32	R:VERB:INFL	0.58	U:VERB:TENSE	0.25
R:PUNCT	2.84	R:MORPH	1.77	M:VERB:TENSE	1.32	R:ADV	0.53	M:PART	0.18
M:PUNCT	2.80	U:PREP	1.69	M:VERB:FORM	1.29	M:NOUN:POSS	0.52	U:PART	0.10
R:VERB	2.71	R:SPELL	1.66	U:NOUN	1.26	R:ADJ	0.51	R:NOUN:POSS	-0.06
R:NOUN	2.64	U:CONJ	1.64	M:NOUN	1.22	M:PRON	0.49	U:PRON	0.06
R:NOUN:NUM	2.32	U:DET	1.62	R:PRON	1.14	R:PART	0.42	U:VERB:FORM	0.05
R:ORTH	2.22	R:WO	1.58	R:VERB:SVA	1.11	R:ADJ:FORM	0.41	M:CONTR	0.02
R:VERB:TENSE	2.08	M:CONJ	1.52	U:CONTR	1.10	R:NOUN:INFL	-0.37	R:CONTR	0.02

Table 14: The standardized regression coefficients of the LR model trained on all the big (55) edit features to measure the impact of each feature on PEET estimation.

time-to-correct value (Section 5.2). We provide the estimated impact of all edit types here.

Model Features	Regression Coefficient
Substitutions (R)	14.05
Deletions (U)	6.71
Insertions (M)	5.28
Sentence Correct (C)	-2.33

Table 15: The standardized regression coefficients of the LR model trained on the small (4) edit features to measure the impact of each feature on PEET estimation.

Model Features	PEET Correlation	Regression Coefficient
# of words in TRG	0.43	14.07
Substitutions (R)	0.47	6.76
# of Edited Words	0.52	6.46
# of Words in MO	0.43	-5.86
Deletions (U)	0.32	3.85
Sentence Correct (C)	-0.3	-2.63
Insertions (M)	0.28	0.66

Table 16: The correlation of the features used to train the small-edits(4) Linear Regression (LR) model in Table 6. We also list the standardized regression coefficients to measure the impact of each feature on PEET estimation.

H PEET Scorer Ranking and Comparison of GEC Tools with Human Judgment Rankings

We evaluate and rank 33 different GEC Tools and correction sets, part of 3 GEC Human Judgment Rankings, to estimate the quality of our PEET Scorer (Section 5.3). We list all the GEC Tools along with the Human Judgment and PEET Scorer rankings here.

Model Name	HJR Score	PEET Score	PEET Ranking
marian	76.99	21.82	1
lstm-r	74.48	22.45	3
lstm	74.3	22.39	2
nus	73.94	22.47	4
transformer	73.9	22.79	5
amu	70.68	23.27	6
input	68.15	23.3	7

Table 17: PEET Scorer estimated average time-to-correct per sentence and ranking for 7 GEC Tool corrections on the FCE dataset (1936 Sentences), along with their Human Judgment Ranking (HJR), presented in *Napoles-FCE* (Napoles et al., 2019) (Section 5.3). The 7 GEC Tools consist of Seq2Seq Neural Models.

Model Name	HJR Score	PEET Score	PEET Ranking
lstm-r	78.27	27.61	2
lstm	77.73	27.61	1
amu	75.98	28.35	5
input	75.89	27.72	3
marian	75.8	30.52	7
nus	75.78	28.34	4
transformer	71.53	29.77	6

Table 18: PEET Scorer estimated average time-to-correct per sentence and ranking for 7 GEC Tool corrections on the WikiEd dataset (1984 Sentences), along with their Human Judgment Ranking (HJR), presented in *Napoles-Wiki* (Napoles et al., 2019) (Section 5.3). The 7 GEC Tools consist of Seq2Seq Neural Models.

Table 17-18 list the estimation scores for the 6 Seq2Seq GEC Tools ranked by Napoles et al. (2019). The chosen models were AMU (Junczys-Dowmunt and Grundkiewicz, 2016), LSTM/LSTM-R (Klein et al., 2018), Marian (Sennrich et al., 2017), NUS (Chollampatt and Ng,

2018), and, Transformer (Vaswani et al., 2017).

Model Name	HJR Score	PEET Score	PEET Ranking
AMU	0.628	25.8	8
RAC	0.566	26.61	13
CAMB	0.561	26.34	11
CUUI	0.55	25.91	9
POST	0.539	26.28	10
UFC	0.513	24.56	2
PKU	0.506	25.63	6
UMC	0.495	25.72	7
IITB	0.485	24.67	3
SJTU	0.463	24.84	4
INPUT	0.456	24.53	1
NTHU	0.437	26.6	12
IPN	0.3	25.62	5

Table 19: PEET Scorer estimated average time-to-correct per sentence and ranking for 12 GEC Tool corrections on the CONLL14 dataset (1312 Sentences), along with their Human Judgment Ranking (HJR), presented in *Grundkiewicz-C14(EW)* (Grundkiewicz et al., 2015) (Section 5.3). The 12 GEC Tools consist primarily of rule-based and statistical machine translation architecture.

Table 19 lists the quality judgment for the 12 GEC Tools that participated in the CONLL14 GEC Task (Ng et al., 2014) performed by Grundkiewicz et al. (2015). AMU (Junczys-Dowmunt and Grundkiewicz, 2014), CAMB (Felice et al., 2014), CUUI (Rozovskaya et al., 2014), IITB (Kunchukuttan et al., 2014), IPN (Hernandez and Calvo, 2014), NARA (Ng et al., 2014), NTHU (Wu et al., 2014), PKU (Zhang and Wang, 2014), POST (Lee and Lee, 2014), RAC (Boroš et al., 2014), SJTU (Wang et al., 2014a), UFC (Gupta, 2014), and UMC (Wang et al., 2014b).

Table 20 lists the recent GEC Tools evaluated by Kobayashi et al. (2024). GPT-3.5 (Coyne et al., 2023), T5 (Rothe et al., 2021), TransGEC (Fang et al., 2023a), BERT-Fuse (Kaneko et al., 2020), Riken-Tohoku (Kiyono et al., 2019), PIE (Awasthi et al., 2019), LM-Critic (Yasunaga et al., 2021), TemplateGEC (Li et al., 2023), GECToR-BERT (Omelianchuk et al., 2020), UEDIN-MS (Grundkiewicz et al., 2019), GECToR-Ens (Tarnavskiy et al., 2022), BART (Lewis et al., 2020).

Model Name	HJR Score	PEET Score	PEET Ranking
REF-F	0.992	30.53	15
GPT-3.5	0.743	26.04	14
T5	0.179	24.37	10
TransGEC	0.175	23.54	3
REF-M	0.067	24.04	8
BERT-Fuse	0.023	23.61	4
Riken-Tohoku	-0.001	23.36	2
PIE	-0.034	23.66	6
LM-Critic	-0.163	24.37	9
Template GEC	-0.168	25.21	13
GECToR-BERT	-0.178	23.78	7
UEDIN-MS	-0.179	23.36	1
GECToR-Ens	-0.234	23.62	5
BART	-0.3	24.75	12
INPUT	-0.992	24.53	11

Table 20: PEET Scorer estimated average time-to-correct per sentence and ranking for 15 GEC Tool corrections on the CONLL14 dataset (1312 Sentences), along with their Human Judgment Ranking (HJR), presented in *SEEDA-C14-All(TS)* (Kobayashi et al., 2024) (Section 5.3). The 15 GEC Tools consist of strong SOA Neural Models.