Estimating Temporal Post-Edit Effort for Grammar Error Correction Tool Evaluation

Anonymous TACL submission

Abstract

Text editing can involve several iterations of revision to improve the grammaticality and fluency of the content. Incorporating an efficient Grammar Error Correction (GEC) tool in the initial editing stage can significantly reduce the time and amount of post-editing (PE) corrections. In this work, we present the first experiment in estimating the Post-Edit Effort in Time (PEET) for a GEC tool based on edit type and sentence structure. We further determine GEC edit types that have the greatest impact on PE temporal effort. Finally, we evaluate PEET as a new metric for assessing the quality of GEC tool performance. As part of this work, we also collect and release a new dataset with PE time and corrections for two GEC tools on the CONLL14 and BEA19 GEC Test sets.

1 Introduction

Text editing is an important part of document writing and can result in several rounds of revisions. 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 prediction to obtain the target correction. Estimating the post-editing (PE) effort required to reach the target correction can be used as a quality estimation metric for the tool.

PE effort for quality estimation has been studied extensively in Machine Translation (MT) (Specia et al., 2009b, 2010). Computer-aided translation systems that utilize tools like Translation Memory (TM) or retrieval (Cai et al., 2021; Xu et al., 2020) also benefit from human-in-the-loop PE. PE in MT is defined as correcting MT outputs by a human translator (Senez, 1998), ensuring that the translated text corresponds to the rules of the target language and is reliable with no inconsistencies in meaning introduced by the MT system. Although some work has explored computer-assisted text editing and error correction systems, indicating that human-in-the-loop editing is beneficial (Basiron, 2012; Dagneaux et al., 1998; Li et al., 2023), PE effort has not been studied extensively for quality estimation of GEC tools.

The usefulness of a GEC tool depends inversely on the PE its predictions require for further correction. PE effort in MT has been studied across three levels (Krings, 2001): technical effort, which is the number of edits; cognitive effort, which denotes the psychological assessment required to identify and correct the errors; temporal effort, which is the total time taken to evaluate and perform post-edits (which includes technical and cognitive effort).

In this work, we explore temporal PE effort for quality estimation of a GEC tool. We design an experiment to evaluate the predictions of two strong GEC tools - GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019) on two popular GEC Test sets - CONLL14 (Ng et al., 2014) and BEA19 (Bryant et al., 2019) by professional human editors. The post-edited target correction and time-to-correct value are used to create a new GEC dataset for the Post Edit Effort in Time (PEET) estimation task for GEC tools. We also train models to estimate PEET (temporal effort) using the count of different edit types (technical effort) and sentence structure features. We further use this dataset to determine edit types that have the greatest impact on PEET. We compare the performance of our PEET estimation model with GEC perceived judgment rankings to explore the effectiveness of PEET-based evaluation of GEC tools. This

is the first work to study the temporal aspect of PE effort for GEC quality estimation.

We make the following contributions:

1. We collect and present the first dataset to study post-editing for GEC, consisting of 3 high-quality target corrections for two GEC Test datasets (BEA19 and CONLL14) - source sentence correction and post-edit for two strong GEC Tools (GECToR and GEC-PD) predictions, along with their respective time-to-correct information.

2. We study the impact of GEC Tools in text-editing, observing editor productivity and efficiency improvement. We also observed that determining whether a sentence needs corrections, paraphrasing, and punctuation changes are the edits that tend to contribute the most to increase PEET.

3. We design models to estimate time-to-correct (PEET) based on the count and type of edits (technical effort), and sentence structure.

4. Finally, we compare our best PEET estimation model with various GEC human judgment rankings, observing its effectiveness for GEC evaluation based on the number and type of corrections (technical effort).

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 models utilize an encoder-decoder architecture where the corrected sentence is generated for each input sentence. The S2S models are often pre-trained with monolingual corpora and synthetic datasets with similar error distribution as the GEC training data (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), which have been fine-tuned on large amounts of human preference and instruction data, also perform well for GEC (Zhang et al., 2023; Fang et al., 2023) 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 predictions made by these models are further corrected by human editors while tracking the time-to-correct (temporal effort). We use this temporal dataset to evaluate the impact of GEC tools for text-editing, observing a reduction in post-editing time and better quality final target 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 4.

2.2 Post Editing Effort and Grammar Error Correction

We briefly review previous work that defines and explores post-editing (PE) effort across three levels (technical, cognitive and temporal effort) (Krings, 2001). Technical effort has been calculated 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 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., 2009a, 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) (Krings, 2001; 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 characteristics (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, similar to our work, Ye et al. (2021) and Tezcan et al. (2019) train models to estimate MT PE time based on edit features. PE has also been explored for Text Summarization Evaluation (Mani et al., 2002) and Natural Language Generation (Sripada et al., 2004).

Technical Effort in PE has been explored for GEC evaluation. GEC metrics and Quality Estimation (QE) methods like ERRANT (Bryant et al., 2017), M^2 (Dahlmeier and Ng, 2012), GoToScorer (Gotou et al., 2020), and GLEU (Courtney et al., 2016) can be considered as Technical Effort estimators as they rely on comparing edits for correction evaluation. Unlike metrics based on edit scores, PT-M^2 (Gong et al., 2022), Scribendi Score (Islam and Magnani, 2021), SOME (Yoshimura et al., 2020) and IMPARA (Maeda et al., 2022) rely on Neural models like BERT (Devlin, 2018) and GPT2 (Radford et al., 2019) for generating sentence representations and perform QE.

Human-assessed judgment rankings of GEC systems (Grundkiewicz et al., 2015; Kobayashi et al., 2024; Napoles et al., 2019), used for evaluation metric assessment, are an estimate of perceived cognitive effort in GEC. However, perceived cognitive effort for PE does not always agree with the actual PE effort (Moorkens et al., 2015; Federico et al., 2014). Similar to MT, some work in GEC has 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 $(0.4, 0.23, 0.16)$. Gotou et al. (2020) proposed a GEC evaluation approach based on edit complexity, calculating complexity based on how many GEC Tools performed the edit. 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 above works in GEC have considered using targeted corrections (the closest correction to a GEC Tool output), which is how PE effort is estimated in MT (like Translation Edit Rate (TER) and Human-TER (Snover et al., 2006)). Chollampatt and Ng (2018) proposed using TER edit score features to estimate the GEC quality of an edit using untargeted references (references that are not the result of correcting a GEC tool output). In their work, the TER score is used interchangeably with the HTER score. However, it has been shown that the TER score correlates poorly with HTER and human-judgment scores (Snover et al., 2006). Apart from estimating the post-editing 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). Rozovskaya and Roth (2021) compared the impact of using targeted/untargeted references for GEC Tool evaluation, generating targeted references for 100 sentences from English and Russian GEC datasets. Similarly, Östling et al. (2023) generated and proposed using post-edited references to evaluate various Swedish GEC systems. In this paper, we present the first work to study the Temporal Effort in Post-Editing for GEC. We also present the first large dataset of post-edited corrections for two strong GEC tools - GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019) predictions on two popular GEC Test sets - CONLL14 (Ng et al., 2014) and BEA19 (Bryant et al., 2019).

3 Dataset Collection and Processing

An important component of our work is creating a high-quality dataset of post-edit corrections for GEC, along with the time-to-correct (temporal effort) required for correction. This work is done in collaboration with a professional text-editing company (Anonymous) who collected this data. This section explains our dataset collection, filtering, and quality estimation process.

3.1 Dataset Source

We use source sentences from two popular GEC Benchmarks - CONLL14 (Ng et al., 2014) and

BEA19 (Bryant et al., 2019) for the evaluation ($1312 + 4477 = 5789$ sentences). Two GEC Tools - GECToR (Omelianchuk et al., 2020) and GEC-PD (Kiyono et al., 2019) (Section 2.1) were used for first-pass source editing. 8 professional text editors, employed by Anonymous, were asked to evaluate and correct the source sentences and post-edit the two GEC Tool output predictions. This resulted in a dataset of 3 target corrections (one for the source + two for the tool prediction corrections) for each of the 5789 sentences (total of $5789 * 3 = 17367$ target corrections; see Appendix A). The time-to-correct was recorded for each sentence.

3.2 Editor Correction Framework

The source sentence and the GEC Tool prediction serve as the basis for the editor's post-editing. This follows the natural setup of Text Editing, since a GEC Tool prediction is evaluated for further correction, compared to the original sentence. The editors were given GEC post-editing (PE) instructions (Appendix D-2) and asked to perform minimal edits and avoid rewrites. We used the Qualtrics[1] survey tool to collect post-edit corrections and used the "Timing Question" feature to log time-to-correct for each source sentence. All other metadata logging was disabled.

The task of evaluating $17,367$ sentences was performed in batches of 50 sentences each. Since our dataset has 3 variations of the same sentence - source sentence as well as GECToR and GEC-PD prediction output, each sentence variation was given to a different professional editor (in a pool of 8 editors). This helped us eliminate any time-to-correct bias. The editors were shown the source sentence and the first-pass GEC Tool prediction output (Appendix D-3). The final target correction and time-to-correct were logged for each sentence. For source sentence post-editing, only the original sentence was presented.

3.3 Data Filtering

Using our collection framework, we collected target corrections for all 3 source sentence variations (Appendix A). To improve the dataset quality, we perform two stages of data filtering on the 3 target correction sets. In the first stage, we eliminate outliers based on the time-to-correct. Snover et al. (2006) showed that their editors took between 3-6 minutes for each correction. Considering this and the distribution of the time-to-correct in this work, we set 250 seconds as the threshold, filtering out corrections with greater time-to-correct values. Finally, we merge duplicate corrections from our dataset by averaging the time-to-correct values. This filtering allows us to retain 81.26% of our dataset that we use as train and test sets (80:20 split). Table 1 contains the dataset size after each filtering stage.

Data Filtering Stages	**Data Size**	**% of Data**
Original Corrections	17367	100%
Time-To-Correct < 250 sec	17033	98.08%
Averaging the Time for Same Corrections	14112	81.26%

Table 1: Dataset size of collected corrections after different filtering stages.

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. Out of these, two are post-edited target corrections after a first-pass edit using GEC Tools, and one is a direct correction of the source sentence. We evaluate the quality of these three target corrections by 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.

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 2 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 these 2 outliers, we observe similar quality scores for our corrections. This indicates that our 3 CONLL14 Target corrections are of high quality. Unfortunately, there are no public correction references available for the BEA19 Test set (this work being the first to

[1] https://www.qualtrics.com/

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 2: The M2 precision and recall quality score for all CONLL14 target correction sets on the official CONLL14 competition task.

present 3 target correction sets), 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 model output prediction on the official BEA19 and CONLL14 competitions and compare trends between the correction sets. We use the BEA19 competition website scorer[2] to evaluate the performance of BEA19 target corrections. Table 3 shows the quality scores for the GECToR and GEC-PD Tool output prediction and the final editor target corrections (EC).

Similar patterns are observed between the CONLL14 and BEA19 target correction sets. We observe a significant increase in Recall scores for the EC compared to the initial GEC Tool prediction 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) and compare the target correction sets for both datasets with themselves to ensure better consistency and quality. The IAA framework proposes that the $F_{0.5}$ multi-reference score, used to evaluate a GECTool-vs-human corrections, can similarly evaluate human-vs-human corrections, and serve as the IAA score. When comparing multiple annotator corrections, a single correction set can be compared against the rest as a reference to get quality scores, the final IAA score being the average of all annotator correction quality 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 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 5 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.

The Avg-IAA and official GEC competition quality scores indicate that the 3 target corrections that we collect for BEA19 and CONLL14 have similar or better quality than other public target corrections.

To ensure we choose strong GEC Tools (Section 2.1) to obtain first-pass output predictions, we compare the quality of the GEC Tool predictions 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 4 show similar performance between the GECToR

[2]BEA19 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 3: 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 4: 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 - $F_{0.5}$
$A1$	$\lvert\{RAND(2) \in \{A-A1\}\rvert = 2$	36.21
$A2$	$\lvert\{RAND(2) \in \{A-A2\}\rvert = 2$	45.48
$A3$	$\lvert\{RAND(2) \in \{A-A3\}\rvert = 2$	46.72
$A4$	$\lvert\{RAND(2) \in \{A-A4\}\rvert = 2$	40.54
$A5$	$\lvert\{RAND(2) \in \{A-A5\}\rvert = 2$	46.01
$A6$	$\lvert\{RAND(2) \in \{A-A6\}\rvert = 2$	50.85
$A7$	$\lvert\{RAND(2) \in \{A-A7\}\rvert = 2$	42.72
$A8$	$\lvert\{RAND(2) \in \{A-A8\}\rvert = 2$	49.46
$A9$	$\lvert\{RAND(2) \in \{A-A9\}\rvert = 2$	52.0
$A10$	$\lvert\{RAND(2) \in \{A-A10\}\rvert = 2$	48.57
Avg-IAA $\{A\}$	$\{A\}, 2$	**45.85**
$c1$	$\lvert\{C-c1\}\rvert = 2$	54.11
$c2$	$\lvert\{C-c2\}\rvert = 2$	57.36
$c3$	$\lvert\{C-c3\}\rvert = 2$	59.14
Avg-IAA $\{C\}$	$\{C\}, 2$	**56.87**
$b1$	$\lvert\{B-b1\}\rvert = 2$	57.94
$b2$	$\lvert\{B-b2\}\rvert = 2$	59.39
$b3$	$\lvert\{B-b3\}\rvert = 2$	59.81
Avg-IAA $\{B\}$	$\{B\}, 2$	**59.05**

Table 5: 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.

and GEC-PD Tool prediction output and the subsequent EC, indicating that GECToR and GEC-PD are strong first-pass GEC Tools.

3.5 Impact of GEC Tools

Comparing the time-to-correct source sentences versus the GEC Tool first-pass predictions, we can

study the impact of using GEC Tools in Text Editing.

EC quality scores presented in Table 3 show that GEC Tool output EC, has better values compared to the Source Sentence EC. In Table 6, we compare the time taken (in seconds) by a human editor to correct the set of BEA19 and CONLL14 sentences with and without first-pass editing by a GEC tool. We observe that GEC Tools also 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 and overall productivity.

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

4 Methodology

In this work, we design models that estimate post-editing effort in time (PEET) for a source sentence given the target correction. We train statistical and neural network (Language Model (LM)) models that can predict the temporal effort in terms of the number and type of edits and sentence structure.

The dataset that we collected contains 3 iterations for each sentence - source (SRC), model (GEC Tool) output (MO) prediction and post-edited target correction (TRG). Different training features in terms of edits and sentence structure can be selected and extracted from a subset of each SRC, MO and TRG triple (Appendix E).

PEET models trained on features selected from $[MO, TRG]$ sentences performed better than models trained on fine-grained features from $[SRC, MO, TRG]$ sentences (Appendix F). Hence, we only discuss the features and results of the model trained using the $[MO, TRG]$ sentences, 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 type. Figure 1 lists the different edit categories and their syntactic type generated by ERRANT.

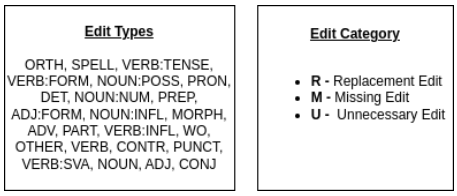

Figure 1: ERRANT edit category and types.

We use the number and type of edits as features for our statistical models. Similar to the edit type 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 18 in Appendix G). We train separate models for all three edit levels (4, 25, 55).

4.2 Sentence Structure Features

Sentence word length and the number of edited words (Technical PE effort) can have an impact on PE effort (Specia, 2011). So, we use the word length of source (Model Output - MO) and target correction sentences as additional training features for the statistical PEET model, while the number of edited words is used as a feature for both statistical and Neural-LM PEET models.

Since semantics and syntax structure have been shown to impact PE effort (Tezcan et al., 2016; Bangalore et al., 2015), we train neural-LM PEET 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). 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.

4.3 PEET Estimation Models

We design Linear Regression (LR) and Support Vector Regression (SVR) PEET models using the ERRANT Edit count and different type levels (4, 25, 55), number of edited words, source sentence length and target corrections as features. For training BERT-Large (Devlin, 2018) and RoBERTa-Large (Liu et al., 2019), we consider 4 different set of features described in Section 4.2 and listed in Table 7. 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 Estimation Models

The results for the Statistical - Linear Regression (LR) and SVR PEET models, 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 8. Results for the neural BERT-Large and RoBERTa-Large PEET models, trained on sentence structure features (Table 7, are presented in Table 9.

The statistical models considering edit type information (25,55 labels) perform better than using minimal substitution, deletion and insertion edit category labels (Figure 1). This indicates that the syntax type of the edit impacts post-editing effort. For neural-LM, flattened syntax tree and part-of-speech features, along with the number of edited words, perform better for BERT, while RoBERTa models trained on sentence embedding and syntax-based features perform better.

Comparing the best statistical and neural PEET estimation models (Table 8 and 9), we observe similar performance in terms of the r-score. We obtain a correlation of $r = 0.565$ for 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 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).

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 10. Coefficients for the other edit granularities (4 labels and 55 labels) and sentence features are provided in Appendix G.

5.3 PEET Model for GEC Quality Estimation

The Post-Edit Effort in Time (PEET) models can utilize sentence and edit type features to estimate the time-to-correct (in seconds) a GEC Tool prediction into the final target. Since an efficient GEC Tool would require lower post-editing (PE) time, the PEET can be used to evaluate the quality of a GEC Tool (Specia, 2011). To determine the feasibility of using a method as a GEC quality estimation tool, the method's ranking is compared to human judgment rankings (HJR) (Section 2.2). However, HJR, which corresponds to human perception (rating) of PE effort, is not always a reliable predictor of the actual post-editing effort (Moorkens et al., 2015).

To observe the correlation between Temporal and perceived PE effort, we compare the PEET-Linear Regression (25 Edit Features) model (Section 4.1) estimation ranking with 3 GEC HJR.

- *Grundkiewicz-C14(EW)* - ranking of 12 GEC systems 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.

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 7: 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.

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 8: 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.

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 9: 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.

- *Napoles-FCE* and *Napoles-Wiki* - ranking of 6 Seq2Seq GEC models on the FCE (Yannakoudakis et al., 2011) and WikiEd (Grundkiewicz and Junczys-Dowmunt, 2014) datasets by Napoles et al. (2019).

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., 2006) method, which tracks relative ranking based on a setwise 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 systems 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 11 shows the Pearson (r) and Spearman (ρ) correlation scores of the HJRs with the PEET

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

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 11: Evaluating 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. Negative correlation indicates a lower time-to-correct score corresponds to a higher human judgment ranking.

model ranking and the Word Error Rate (WER) (Snover et al., 2006), which tracks the number of edits required to correct a GEC Tool prediction.

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 dependent on WER scores (Technical PE effort - Section 2.2). We also observe a better alignment of WER and PEET with the HJRs collected using the PRWS method, rather than TS or EW.

These results suggest that our PEET model can be an effective GEC evaluation tool when the GEC Tool prediction quality depends on the Technical Post-Editing Effort (WER and type of edits) required to obtain the final correction. However, the PEET model does not provide a reliable GEC quality estimation tool when the evaluation is dependent on the perceived PE efforts.

6 Conclusion

In this work, we perform the first study on Post-Edit Effort Estimation in Time (PEET) for the task of Grammar Error Correction (GEC). As part of this study, we collect and present the first high-quality large dataset of GEC post-edit corrections along with their respective time-to-correct values. We also analyze this data to study the impact of GEC Tools in Text Editing, observing an improvement in editor efficiency and productivity.

We extract various automated sentence structure and edit category features from the source-correction sentence pairs to train models that estimate the PEET for a GEC Tool prediction. Comparing the impact of different edit types on the post-editing (PE) effort, we find that rewrites, paraphrases and modifying punctuation had the highest impact on PEET. It was also observed that determining whether a sentence is correct or not is a time-consuming task that significantly impacts time-to-correct values.

Finally, we observe that our PEET model can be an effective GEC evaluation tool when the correction quality is dependent on the Technical PE Effort (type and amount of edits). However, similar to work in Machine Translation, it is inconsistent with quality estimation based on perceived PE efforts.

7 Future Work and Limitations

Since we present the first study and dataset of Post-Editing Effort in Time for GEC, our goal is to provide a baseline for future work in this area. An interesting area of future work is exploring post-editing features to train stronger PEET models. Our work is currently limited to features generated by the ERRANT toolkit (Bryant et al., 2017). Recently, there has been some work in the area of Grammar Error Explanation to define and predict 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). It would be interesting to explore these descriptive edits as features for the PEET model.

Evaluating our PEET model as a GEC quality estimation tool shows that it is effective when the correction quality is dependent on the technical post-editing effort and not perceived cognitive effort. Studying actual cognitive effort for GEC post-editing and comparing it with technical and temporal effort is another interesting direction for future work.

Finally, we acknowledge that our work is limited to the English language. Future work on post-editing GEC for other languages can show the impact of language type on PEET for GEC.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. *arXiv preprint arXiv:1910.02893*.

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, et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Halizah Basiron. 2012. *A Statistical Model of Error Correction for Computer Assisted Language*

Learning Systems. Ph.D. thesis, University of Otago.

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*.

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.

Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. Neural machine translation with monolingual translation memory. *arXiv preprint arXiv:2105.11269*.

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 Yeoil 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. Neural quality estimation of grammatical error correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2528–2539.

Napoles Courtney, Sakaguchi Keisuke, Post Matt, R Tetreault Joel, et al. 2016. Gleu without tuning. *arXiv*.

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 Hartsuker, 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*.

Estelle Dagneaux, Sharon Denness, and Sylviane Granger. 1998. Computer-aided error analysis. *System*, 26(2):163–174.

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. *arXiv preprint arXiv:2104.04986*.

Christopher Davis, Andrew Caines, Øistein Andersen, Shiva Taslimipoor, 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. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Marcello Federico, Matteo Negri, Luisa Bentivogli, and Marco Turchi. 2014. Assessing the impact of translation errors on machine translation quality with mixed-effects models. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1643–1653.

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.

Peiyuan Gong, Xuebo Liu, Heyan Huang, and Min Zhang. 2022. Revisiting grammatical error correction evaluation and beyond. *arXiv preprint arXiv:2211.01635*.

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.

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.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: a bayesian skill rating system. *Advances in neural information processing systems*, 19.

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.

Anisia Katinskaia and Roman Yangarber. 2024. Gpt-3.5 for grammatical error correction. *arXiv preprint arXiv:2405.08469*.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. *arXiv preprint arXiv:1805.01052*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. *arXiv preprint arXiv:1909.00502*.

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, Wilker Aziz, Luciana Ramos, Lucia Specia, Jussi Rautio, Lauri Carlson, Inari Listenmaa, Seppo Nyrkkö, Gorka Labaka, Arantza Díaz De Ilarraza, et al. 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.

Hans P Krings. 2001. *Repairing texts: Empirical investigations of machine translation post-editing processes*, volume 5. Kent State University Press.

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.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. 2002. Summac: a text summarization evaluation. *Natural Language Engineering*, 8(1):43–68.

Joss Moorkens, Sharon O'brien, Igor AL Da Silva, Norma B de Lima Fonseca, and Fabio Alves. 2015. Correlations of perceived post-editing effort with measurements of actual effort. *Machine Translation*, 29:267–284.

Maria Nadejde and Joel Tetreault. 2020. Personalizing grammatical error correction: Adaptation to proficiency level and l1. *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 Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

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. *arXiv preprint arXiv:2404.14914*.

Robert Östling, Katarina Gillholm, Murathan Kurfalı, Marie Mattson, and Mats Wirén. 2023. Evaluation of really good grammatical error correction. *arXiv preprint arXiv:2308.08982*.

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*.

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*.

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.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. Coedit: Text editing by task-specific instruction tuning. *arXiv preprint arXiv:2305.09857*.

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.

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*.

Matthew Snover, Bonnie Dorr, Richard 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.

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*.

Lucia Specia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *Proceedings of Machine Translation Summit XIII: Papers*.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24:39–50.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009a. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual conference of the European Association for Machine Translation*.

Lucia Specia, Marco Turqui, Zhuoran Wang, John Shawe-Taylor, and Craig Saunders. 2009b. Improving the confidence of machine translation quality estimates. In *Proceedings of Machine Translation Summit XII: Papers*.

Somayajulu G Sripada, Ehud Reiter, and Lezan Hawizy. 2004. Evaluating an nlg system using post-editing. *WEATHER*, 5(7).

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.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. *arXiv preprint arXiv:2203.13064*.

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, Véronique Hoste, and Lieve Macken. 2016. Detecting grammatical errors in machine translation output using dependency parsing and treebank querying. *Baltic Journal of Modern Computing*, 4(2):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, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Jitao Xu, Josep-Maria Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579. Association for Computational Linguistics.

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.

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 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.

Zheng Yuan, Shiva Taslimipoor, 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.

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 PEET Dataset Distribution

Sentence Set Type	CONLL14 Test Set	BEA19 Test Set	Total
Source Sentence	1312	4477	5789
GECToR Correction	1312	4477	5789
GEC-PD Correction	1312	4477	5789
Total	3936	13431	17367

Table 12: Distribution of human editor post-edit target correction counts by dataset and source type. We collect and present a dataset of 17367 Target corrections.

B GEC Evaluation File Example and Format

The evaluation of a GEC Model requires a Source (S), Target (T) and Model Output (MO) sentence. Table 13 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 :	Surrounded by such concerns, it is very likely that we are distracted to worry about these problems.
Target :	Surrounded by such concerns, it is very likely that we will be too distracted to worry about these problems.
Model Output :	Surrounded by such concerns, it is very likely that we are distracted from worrying about these problems.

Table 13: *Source*, *Target* and example *Model Output* made by a GEC Model.

• *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 generated an alignment and detection of 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, but added the use of syntactic and linguistic features in text 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 library[3]. 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 $alpha = 1.0$.

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 stronger neural predictive models, we fine-tuned the BERT-Large (Devlin, 2018) 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 library[4] 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 Huggingface[5] training pipeline.

D GEC Post Editing Instructions and Survey Example

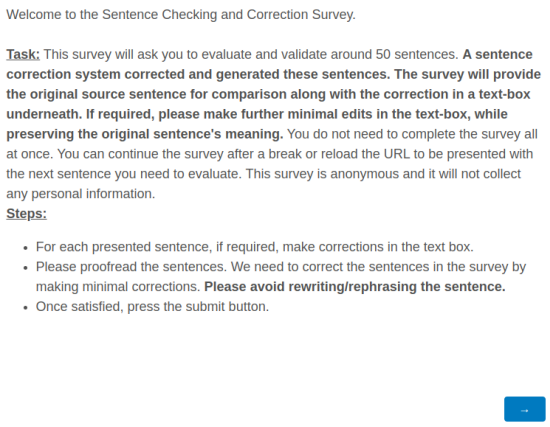

Figure 2: Survey instructions for the editor to perform post editing, and obtain target corrections for our dataset.

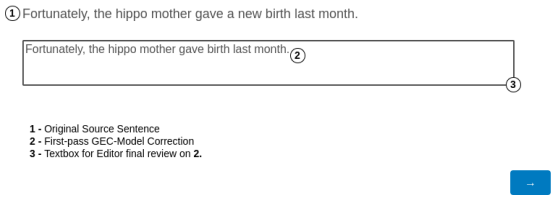

Figure 3: Example source sentence and its first-pass edit from the Survey. The editor can make further improvements in the text box. Submitting the final target correction.

E Different Sources for Training Feature Selection and Extraction

Our dataset has 3 iterations for each sentence. We have the original sentence - source (SRC), the first-pass correction by a GEC Tool - Model Output (MO) and the final targeted editor correction - target (TRG). Out of our 3 correction sets, 2 sets have a first-pass correction performed by a GEC Tool, and 1 set has the correction for the source sentence. For source sentence correction, we consider the Model Output to be the Source Sentence ($MO = SRC$). This can be summarized as using a GEC Tool that makes 0 edits to a source.

Figure 4 shows the editing 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:

(a) MO (b) SRC - MO (c) **MO - TRG** (d) **SRC - MO - TRG**

Strong post-editing features can be extracted from the $SRC - MO - TRG$ and $MO - TRG$ sentence

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
[4] https://adapterhub.ml/
[5] https://huggingface.co/

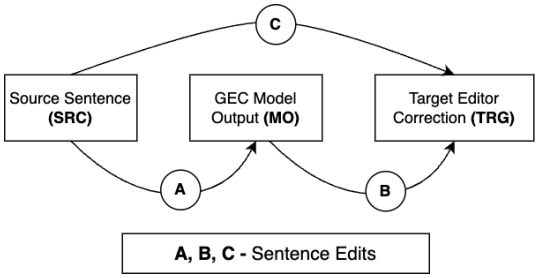

Figure 4: Sentence correction edits extracted using the ERRANT toolkit.

pairings. The source sentence can be further used to divide correction edits into ignored and incorrect edits.

- **SRC - MO - TRG**: We consider and extract the set of edits - A and C (Figure 4) 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 4) as the input for the trained models.

We found that the performance of models trained on these 2 feature sources was comparable (Appendix F). This also indicates that the model can get most of the information from the post-editing correction stage - B. In our work, we only present and discuss the results of the model trained using the $MO - TRG$ sentence features.

F PEET Models using SRC, MO and TRG Sentence Features

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

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

Table 15: Neural PEET model performance over 5 runs using the source (SRC), GEC Model Output (MO) and Target Correction (TRG) sentence features. The results are shown as Pearson Correlation (r) and Mean Absolute Error (MAE) loss.

G Feature Impact on Post-Editing Time using Regression Coefficients

Model Features	PEET - r	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 8. We also list the standardized regression coefficients of the LR model to measure the impact of each effort on PEET estimation.

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

Table 17: The standardized regression coefficients of the LR model trained on the small (4) edit features to measure the impact of each effort on PEET estimation.

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