Towards Automated Document Revision: Grammatical Error Correction, Fluency Edits, and Beyond

Masato Mita[1,2] **Keisuke Sakaguchi**[3] **Masato Hagiwara**[4,5]
Tomoya Mizumoto[2] **Jun Suzuki**[3,2] **Kentaro Inui**[6,3,2]
[1]CyberAgent [2]RIKEN AIP [3]Tohoku University
[4]Earth Species Project [5]Octanove Labs [6]MBZUAI

Abstract

Natural language processing (NLP) technology has rapidly improved automated grammatical error correction (GEC) tasks, and the GEC community has begun to explore *document-level* revision. However, there are two major obstacles to going beyond automated *sentence-level* GEC to NLP-based *document-level* revision support: (1) there are few public corpora with document-level revisions annotated by professional editors, and (2) it is infeasible to obtain all possible references and evaluate revision quality using such references because there are infinite revision possibilities. To address these challenges, this paper proposes a new document revision corpus, **T**ext **R**evision of **A**CL papers (TETRA), in which multiple professional editors have revised academic papers sampled from the ACL anthology. This corpus enables us to focus on document-level and paragraph-level edits, such as edits related to coherence and consistency. Additionally, as a case study using the TETRA corpus, we investigate reference-less and interpretable methods for meta-evaluation to detect quality improvements according to document revisions. We show the uniqueness of TETRA compared with existing document revision corpora and demonstrate that a fine-tuned pre-trained language model can discriminate the quality of documents after revision even when the difference is subtle.

1 Introduction

Document revision is a crucial step in the process of writing essays and argumentative texts. The writing process consists of two major parts: content organization and selection planning (henceforth, *planning part*) and realization of text improvement (henceforth, *realization part*), which are hierarchical and recursive. In addition, according to previous studies on argumentative writing (Flower and Hayes, 1981; Beason, 1993; Buchman et al., 2000; Seow, 2002; Allal et al., 2004), *realization part* in writing process typically comprises three main stages: *Revising*, *Editing*, and *Proofreading*. *Revising* is the initial editing step used to plan and structure the overall document at a high level, *Editing* focuses on making sentence-level or phrase-level expressions, and *Proofreading* is used to identify and correct errors such as spelling and grammar errors (see Figure 1, left). While the order of these steps is not set in stone, the writing process typically starts with a broad, high-level perspective, and gradually narrows down the scope of edits.

In contrast to the typical human writing process, GEC research in NLP field, which is primarily intended to support writing, initially focused on a fine-grained scope, e.g., spelling errors (Brill and Moore, 2000; Toutanova and Moore, 2002; Islam and Inkpen, 2009) and closed-class parts of speech (such as prepositions and determiners) (Han et al., 2006; Nagata et al., 2006; Felice and Pulman, 2008). The research community then expanded its focus to include edits at the phrase and sentence levels while also considering fluency (Sakaguchi et al., 2016; Napoles et al., 2017) (Figure 1, right). However, significantly less work has been done on *document-level* revisions due to two major challenges. First, document revisions encompass a broader range of concerns such as coherence and flow, compared to conventional GEC and fluency correction, which makes it difficult to find publicly available corpora that have been annotated by experts (professional editors). Second, evaluating the quality of revisions is challenging as it requires multiple reference points, as there are many ways to revise a single document. This suggests that *reference-less* evaluation metrics (Napoles et al., 2016; Choshen and Abend, 2018; Islam and Magnani, 2021) are hold significant importance in automated document revision models.

Considering these challenges associated with automated document revision, we propose a new high-quality corpus and explore possibilities for transpar-

Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, pages 251–265
June 20, 2024 ©2024 Association for Computational Linguistics

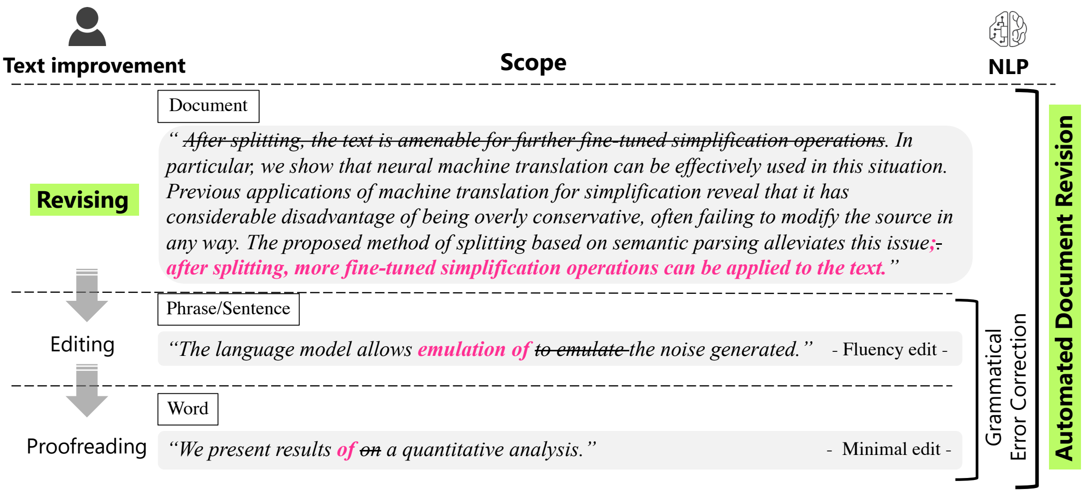

Text improvement
Scope
NLP
Document
" ~~After splitting, the text is amenable for further fine-tuned simplification operations~~. In particular, we show that neural machine translation can be effectively used in this situation. Previous applications of machine translation for simplification reveal that it has considerable disadvantage of being overly conservative, often failing to modify the source in any way. The proposed method of splitting based on semantic parsing alleviates this issue;~~.~~ after splitting, more fine-tuned simplification operations can be applied to the text."
Revising
Phrase/Sentence
"The language model allows emulation of ~~to emulate~~ the noise generated."
- Fluency edit -
Editing
Word
"We present results of ~~on~~ a quantitative analysis."
- Minimal edit -
Proofreading
Grammatical Error Correction
Automated Document Revision

Figure 1: Overview of the scope for automated document revision. Each example is taken from TETRA corpus. We focus on the document revision process which has been overlooked by GEC. Automated document revision extends the scope of GEC.

ent evaluation methods that are independent of gold standards or references. Our corpus, **T**ext **R**evision of **ACL** papers (TETRA), comprises academic papers from the ACL anthology with document-level revisions, revision types, and concrete feedback comments annotated by multiple professional editors. This corpus was designed based on a new XML-based annotation scheme that can handle edit types beyond sentences (e.g., argument flow) in addition to conventional word-level and phrase-level edits. TETRA has uniqueness in terms of the number of references, the expertise level of the editors, and topic diversity.

As a case study, we use TETRA to investigate whether it is possible to build an **i**nstance-wise **r**evision **c**lassification (IRC) method, in which a model can distinguish pre-edited or post-edited versions for a given single revision pair. In recent years, several studies have been conducted on the use of large language models (LLMs) as evaluators in language generation tasks. For example, GPT-4 (OpenAI, 2023) has demonstrated superior performance compared to existing automatic evaluation metrics in text summarization, dialogue generation, and machine translation (Liu et al., 2023; Kocmi and Federmann, 2023). In light of this current situation, we conduct experiments to evaluate how well pre-trained language models, such as BERT (Devlin et al., 2019) and LLMs such as GPT-4, can perform as a (meta-)evaluation method for each edit type, both with and without fine-tuning. The results demonstrate that the supervised method can accurately choose post-edited snippets with an accuracy of 0.85 to 0.96, indicating the feasible potential of automated evaluation in document revision.

We release TETRA to the public, and hope that it will encourage the community to work towards automated document-level revision.[1]

2 Background

The field of GEC, which has a multi-decade history, began with the goal of detecting and correcting targeted error types and providing feedback to English as a second language learners.[2] Early GEC systems primarily focused on a limited number of closed-class error types, such as articles (Han et al., 2006) and prepositions (Chodorow et al., 2007; Tetreault and Chodorow, 2008; Tetreault et al., 2010; Cahill et al., 2013; Nagata et al., 2014). The scope of GEC was later expanded to include all types of errors, including verb forms, subject-verb agreement, and word choice errors (Lee and Seneff, 2008; Tajiri et al., 2012; Rozovskaya and Roth, 2014). This line of research led to the establishment of shared benchmark tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014).

Motivated by the observation that error-coded local edits do not always sound natural to native speakers, the scope of GEC has been further expanded from word-level closed-class edits to phrase-level and sentence-level *fluency* ed-

[1] https://github.com/chemicaltree/tetra

[2] In this paper, we focus on GEC literature after the 2000s when statistical were widely adopted. For a comprehensive history of GEC in the 1980s and 1990s, including rule-based approaches, please refer to Leacock et al. (2014).

🟥 Grammaticality 🟪 Fluency 🟨 Clarity 🟩 Style 🟧 Readability 🟪 Redundancy 🟦 Consistency
This paper presents empirical studies and closely corresponding theoretical models of a chart parser's performance while~~the performance of a chart parser~~ exhaustively parsing the Penn Treebank with the Treebank's own context-free grammar (CFG)~~CFG grammar~~. We show how performance is dramatically affected by rule representation and tree transformations, but little by top-down vs. bottom-up strategies. We discuss grammatical saturation, provide an~~, including~~ analysis of the strongly connected components of the phrasal nonterminals in the Treebank, and model how, as sentence length increases, regions of the grammar are unlocked, increasing the effective grammar rule size ~~increases as regions of the grammar are unlocked,~~ and yielding super-cubic observed time behavior in some configurations.
We expect this approach to yield the following three improvements. Taking advantage of the representation learned by the English model will lead to shorter training times compared to training from scratch. Relatedly, the model trained using transfer learning will require ~~requires~~ less data for an equivalent score than a German-only model. Finally, the more layers we freeze the fewer layers we will need to back-propagate through during training; thus,~~. Thus~~ we expect to see a decrease in GPU memory usage since we do not have to maintain gradients for all layers.
We present the results of ~~on~~ a quantitative analysis of a number of publications in the NLP domain on the collection~~collecting~~, publishing, and availability of research data. We find that, although a wide range of publications rely on data crawled from the web, ~~but~~ few publications provide~~give~~ details of~~on~~ how potentially sensitive data was treated. In addition ~~Additionally, we find that~~, while links to repositories of data are given, they often do not work, even a short time after publication. We present~~put together~~ several suggestions on how to improve this situation based on publications from the NLP domain, as well as ~~but~~ also other research areas.

Table 1: Examples of revision. Each edit type is highlighted respectively.

its (Sakaguchi et al., 2016). With this expansion, the community has proposed new benchmark datasets (Daudaravicius et al., 2016; Napoles et al., 2017; Bryant et al., 2019; Napoles et al., 2019; Flachs et al., 2020; Zhang et al., 2023) and evaluation metrics (Dahlmeier and Ng, 2012; Felice and Briscoe, 2015; Napoles et al., 2015; Bryant et al., 2017; Napoles et al., 2019; Gotou et al., 2020; Gong et al., 2022; Ye et al., 2023) for sentence-to-sentence GEC. In addition, GEC models with deep neural network (DNN) techniques have been developed. Such models are robust against word-level and phrase-level local edits in a given sentence and exhibit human-parity performance on some benchmark datasets (Yuan and Briscoe, 2016; Ji et al., 2017; Chollampatt and Ng, 2018; Ge et al., 2018; Kiyono et al., 2019; Kaneko et al., 2020; Rothe et al., 2021; Li et al., 2023; Yang et al., 2023; Fang et al., 2023; Cao et al., 2023).

In contrast to the significant advancements in the area of grammar and fluency correction, relatively few studies have explored revisions for *document-level argumentative writing*, which require a greater investment of time and resources to create appropriate corpora or datasets. Lee and Webster (2012) made an initial attempt to construct a document revision corpus comprising 13,000 student writings with feedback comments from tutors in the Teaching English to Speakers of Other Languages (TESOL) program. Although the authors developed labels for paragraph-level revisions (e.g., coherence), only 3% of all revisions were annotated as paragraph-level revisions, 90% of the revisions were at the word-level, and 7% were at the sentence-level. This is because the corpus comprises writing from language learners, and the majority of errors were simple grammar and fluency errors. This lesson highlights the importance of using a corpus for document-level revision that has already been partially edited for grammar and fluency. However, due to copyright restrictions, this corpus may not be publicly available. The data source for a document-level corpus should be openly licensed to encourage community-based open research in the long term.

Another line of work (Zhang and Litman, 2014, 2015; Zhang et al., 2016, 2017; Kashefi et al., 2022) has created the ArgRewrite corpus, a collection of 86 argumentative essays that include three drafts, each with two cycles of revisions, and edit labels. The ArgRewrite corpus (both v1 and v2) contains roughly half of all edits as surface-level corrections (e.g., conventional GEC or fluency edits), and the other half of edits as content-level document revisions. While the ArgRewrite corpus has more document-level revisions than the corpus of Lee and Webster (2012), all of the essays in the ArgRewrite corpus were written on the same topic. The first version of the ArgRewrite corpus (Zhang et al., 2017) discusses the topic of *whether the proliferation of electronic enriches or hinders the development of interpersonal relationships*, and the second version (Kashefi et al., 2022) focuses on *whether to support or against self-driving cars*.

	Lee and Webster (2012)	Zhang et al. (2017)	Kashefi et al. (2022)	Du et al. (2022)	Ours (TETRA)
# docs	3,760	60	86	559	64
# sents (avg)	-	18.7	25.8	7.19	**26.92**
# references	1	1	1	1	**3**
Edit scope	Form?	Content&Form	Content&Form	Content&Form	**Form**
% beyondGECs	3.2	49.4	52.6	52.8	**56.9**
Drafted by	ESL	Native (*ESL)	Native (*ESL)	Native (*ESL)	**ESL/Native**
Revised by	Author (NonExp.)	Author (NonExp.)	Author (NonExp.)	Author (NonExp.)	**Exp.**
Edit-types by	NonExp.	NonExp.	NonExp.	NonExp.	**Exp.**
Feedback					✓
Topic diversity	✓			✓	✓
Public availability		✓	✓	✓	✓

Table 2: Characteristics of TETRA corpus compared to existing document revision corpora. The **uniqueness of TETRA** is highlighted. *Exp.* and *NonExp.* means expert and non-expert, respectively. *Edit scope* indicates whether it includes edits regarding content and/or form. *% beyondGECs* shows the ratio of edits that are not covered by GEC edit types. *Drafted by* indicates who wrote the (first) draft, *Revised by* shows who revised the draft, *Edit-types by* shows who annotates edit types. *Feedback* (✓) presents whether the corpus contains feedback comments or not. *Topic diversity* (✓) presents whether the corpus contains two or more topics, or a single topic only (no ✓). *Public availability* (✓) shows whether the corpus is publicly available to the community. Native (*ESL) indicates that most of the documents are drafted by native speakers, but some ESL is included.

This lack of topic diversity can lead to overfitting when developing and evaluating automated document revision models (Mita et al., 2019).

Recently, Du et al. (2022) released a corpus of iterative document revisions from Wikipedia, arXiv, and Wikinews, with edit intention labels annotated[3]. Although this work shares the same objective as ours, there are some differences such as the revision scope, the number of references, the expertise level of the editors, and the absence of feedback comments (Table 2). Furthermore, their annotations are done at a sentence level, whereas our dataset (TETRA) is annotated at a document (and sentence) level. Therefore, our dataset (TETRA) complements their corpus (and vice versa).

3 Automated Document Revision

Given a source document d that consists of paragraphs, a potentially automated editor f revises (R) d into d' ($f : d \mapsto d'$). Here, revision R is a set of edits e, and an edit e is defined as a tuple $e = (src, tgt, t, c)$, where *src* is the source phrase before the revision, *tgt* is the revised phrase, t is the edit type (e.g., grammar, word choice, or consistency), and c represents (optional) rational comments about the edit. When *src* is empty (Ø), this edit indicates *insertion*, and it indicates *deletion* when *tgt* is empty; otherwise, the edit is considered to be a *substitution*. Automated document revision includes various edit types (t), e.g., mechanics, word choice, conciseness, and coherence. This is discussed in further detail in §4.4. Note that t does not exclude the scope of conventional (sentential and subsentential) grammatical error and fluency correction. Rationale comments (c) are a useful resource in the study of feedback generation, which has become prominent in the GEC community (Nagata, 2019; Hanawa et al., 2021; Nagata et al., 2021). Thus, automated document revision is a natural extension of sentence-level error correction to document-level error correction with a wider context.

4 The TETRA Corpus

The validity of a dataset design is contingent upon the purpose and goals of the study. In line with §1 (and also Figure 1), the primary objective of this study is to introduce a novel task focused on enhancing document-level editing and its automated evaluation technologies, which is distinct from the existing GEC task. It is important to note that our aim is not to contribute to a broader understanding of "human revision" in general, which sets our study apart from the previous studies on revision (mentioned in §2).. Hence, it is crucial to create a dataset that minimizes the inclusion of minor grammatical errors and fluency-related edits, which are already emphasized as requirements in GEC. This is essential because proposing a new task entails the need to distinguish the technological aspects and linguistic phenomena targeted by the existing task and the proposed task.

[3] We are aware that other subsequent studies (Jiang et al., 2022; D'Arcy et al., 2023) and on text revision have appeared since the preprint of this study was published.

Aspects	Edit types (abr.)	Definition	Scope	%
Grammaticality	grammar, capitalization	edits that aimed to fix spelling/grammar mistakes	S	19.4
Fluency	word choice, word order	edits that aimed to increase sentence fluency	S	23.7
Clarity	clarity	edits that aimed to amplify meaning for clarity	S/D	19.4
Style	style, tone	edits that aimed to adapt the style	S/D	8.0
Readability	readability	edits that aimed to improve readability	S/D	16.8
Redundancy	redundancy, conciseness	edits that aimed to reduce redundancy	S/D	7.2
Consistency	consistency, flow	edits that aimed to increase paragraph fluency	D	5.5

Table 3: Definition of edit types. S and D (in the *scope* column) indicate the sentence and the document, respectively. We highlight edit types that rely on beyond sentence-level context to edit.

4.1 Data Source

To meet the aforementioned requirement, we utilized the ACL anthology [4] papers as our source data. These papers are generally well-written, peer-reviewed papers on NLP. This choice was made based on the hypothesis that addressing minor errors, such as grammatical errors, is necessary to observe global edits that improve coherence and consistency. Furthermore, (2) we chose the abstract and introduction sections since these sections tend to contain fewer embedded math and complex citations than other sections , and they are more likely to induce global editing specific to the document level due to their greater linguistic freedom.

We selected the source documents from the ACL anthology as follows. First, we created eight groups (=2^3) based on the possible combinations of three different attributes: (1) whether the paper was published at a conference or a workshop, (2) whether the paper is affiliated with a native vs. non-native English speaking country, and (3) whether the first author was a student (at the time the paper was published). We randomly sampled papers until we obtained eight unique papers for each group (i.e., 64 papers in total).

4.2 Annotation Scheme

The scope and granularity of edit types vary widely in previous studies, and there is no standard set of labels. Thus, we define categories of edit types (Table 3) based on previous literature on argumentative and discourse writing (Kneupper, 1978; Faigley and Witte, 1981; Burstein et al., 2003; Zhang et al., 2017). Table 1 provides concrete examples of each type of edit in TETRA.

To create the proposed TETRA, we selected an XML format for the following reasons. First, XML is easy to parse using standard libraries (e.g., Python ElementTree and the Java DOM parser)[5] compared to other formats that frequently require exclusive scripts. Such exclusive scripts incur higher maintenance costs to keep up with the updates of additional dependencies. Second, XML is more flexible than other formats in terms of embedding additional information, such as edit types, edit rationale, comments, and other meta information. For example, as shown in Table 1, document revisions include edit types based on various evaluation aspects, and can be further annotated for each edit with their rational comments using a flexible XML scheme (See Appendix C). Furthermore, edits beyond a single sentence, including sentence merging, splitting, and reordering, can be annotated in a flexible manner (See lines 5-7 in Table 7).

4.3 Annotators

We recruited three professional editors with years of experience editing and proofreading English academic writing, who are native English speakers, to independently revise all 64 documents on the Google Docs platform. They added an edit rationale whenever appropriate, and the revised documents were converted to XML format by the first two authors.[6] Information on how to recruit annotators and instructions for them can be found in the Appendix A and B, respectively.

4.4 Statistical Analysis

Table 2 summarizes the characteristics of TETRA corpus compared to existing document revision corpora. We can first emphasize the quality of the TETRA corpus since it is the only document

[4] https://aclanthology.org

[5] We made the nest of XML tags as shallow as possible for users to parse documents even more easily. In TETRA, the maximum depth of nested XML tags is two. We have established an annotation policy for cases of intersecting edit spans, but we did not encounter any such cases made by professional editors.

[6] During the conversion process, minor corrections and remapping of edit types were made only as necessary.

	Student		Non-student		Native		Non-native		Conf.		WS	
Aspects	#	%	#	%	#	%	#	%	#	%	#	%
Grammaticality	79	19.5	106	21.5	60	16.5	125	21.3	110	22.7	75	16.2
Fluency	115	25.2	110	22.4	74	20.4	151	25.8	99	20.4	126	27
Clarity	100	21.9	84	17.1	88	24.2	96	16.4	84	17.3	100	21.6
Style	39	8.5	37	7.5	29	8.0	47	8.0	46	9.5	30	6.5
Readability	74	16.2	85	17.3	75	20.7	84	14.3	92	19.0	67	14.4
Redundancy	32	7.0	36	7.3	22	6.1	46	7.8	25	5.2	43	9.3
Consistency	18	3.9	34	6.9	15	4.1	37	6.3	29	6.0	23	5.0

Table 4: Distributions of revision aspects by writer's attributes.

Levels	Avg	Min	Max
detection	0.32	0.27	0.35
correction	0.83	0.75	1.00

Table 5: Two levels of inter-annotator agreement: agreement on *detection* and *correction*.

revision corpus that is annotated with revisions by multiple experts, whereas most existing document revision corpora are based on revisions by authors themselves, leaving the quality of revisions in doubt. Existing corpora also have the limitation that the editor (*Revised by*) and the edit type annotator (*Edit-type by*) do not coincide, and thus cannot fully reflect the edit intent, but TETRA corpus overcomes this limitation since the edit type is provided by the person who made the revision. Furthermore, we find that the TETRA corpus contains more edits beyond the GEC (*% beyondGECs*) than the existing corpora, indicating that our hypothesis in source data selection (§4.1) is valid.

The right-most column in Table 3 shows the distribution of edit types found in 16 randomly sampled papers (i.e., 25% of the proposed TETRA corpus). We found that 56.9% of the edits were related to issues beyond the sentence-level context (e.g., redundancy), which is greater than other document revision corpora (Table 2). This is simply because TETRA's source documents are academic papers that have already been proofread to some degree compared to other existing document revision corpora where language learner essays are used as the source material. In terms of the differences among the three different attributes (§ 4.1), we did not find any clear trends, which indicates that the quality of papers in the ACL corpus is uniformly good across the venue and author attributes. The details are shown Table 4.

In document-level revision, it is not straightforward to compute inter-annotator agreement due to the diversity of potential revisions and the broad scope of applicable edits. Thus, we measured two levels of inter-annotator agreement, i.e., (1) agreement on *detection* and (2) agreement on *correction*. The first measurement computes how frequently edit spans overlap (i.e., agree) among annotators, and the second measurement computes how frequently edit type labels (e.g., clarity) match when two or more annotators detect the same (or overlapped) span. Table 5 shows the results.

The result demonstrates that the expert annotators agreed on the direction of editing when they decided an issue was in a certain span (the agreement rate on *correction* was approximately 0.8); however, the experts disagreed on where to consider an issue (the agreement rate on *detection* was approximately 0.3), which is a unique characteristic of automated document revision that differs from traditional GECs.

5 A Case Study: (Meta) evaluation

In addition to creating a corpus for automated document revision, it is essential to establish an evaluation that can measure a document's quality improvement (and possibly deterioration) relative to the applied revisions. As a case study, we use TETRA to investigate reference-less and interpretable methods for a (meta-)evaluation method to detect quality improvements according to document revisions.

5.1 How do we evaluate revisions?

Ultimately, the evaluation of document revision systems itself is a research challenge that could be as difficult as building high-quality automated essay scoring (AES) systems (Dikli, 2006). A typical scenario for evaluating text generation is to compute the textual similarity between the hypothesis and references, as in machine translation (BLEU (Papineni et al., 2002)) and summarization

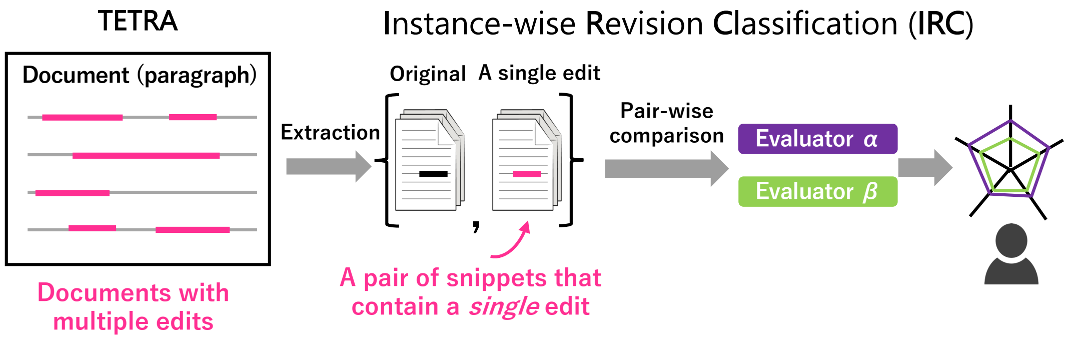

TETRA
Document (paragraph)
Documents with multiple edits
Extraction
Instance-wise Revision Classification (IRC)
Original
A single edit
A pair of snippets that contain a *single* edit
Pair-wise comparison
Evaluator α
Evaluator β

Figure 2: Overview of the IRC meta-evaluation with TETRA.

(ROUGE (Lin, 2004)). However, it is infeasible to elicit all possible gold references for document revision because there are infinite ways to edit a document. In fact, existing work using BLEU and ROUGE to evaluate document revisions shows that such reference-based metrics do not work due to the limited gold references (Du et al., 2022). In addition, given that the purpose of document revision is to support writing, simply presenting users (e.g., model developers and authors) with a single number (overall score) would be insufficient in terms of interpretability and transparency.

In light of the above, a good starting point for a first evaluation method for document revisions would be to develop an explanatory reference-free evaluation model for each evaluation perspective (e.g., clarity, readability, consistency) and then conduct a multidimensional evaluation using this model in an integrated manner.

5.2 Instance-wise revision classification

When using reference-free evaluation as described in §5.1, it is necessary to conduct a *meta-evaluation* of automatic evaluation models (evaluators) to see how well they correlate with human judgments and how reliable they are. Here, it is difficult to measure the quality of a revision automatically based on an *absolute* metric because a single document will contain a variety of edits based on many aspects of evaluation (Table 3). Thus, it is more straightforward to consider a *relative* metric, where a pair of documents is subject to a binary classification choosing the revised one. Such a pairwise comparison has been proven effective as a meta-evaluation method in cases where absolute evaluation is difficult (Guzmán et al., 2015; Christiano et al., 2017). Also, note that document revision contains multiple edits; thus, the binary prediction process cannot identify which edit(s) contributed to the improvement or the degree of improvement.

To address these concerns, we present **I**nstance-wise **r**evision **c**lassification (IRC) as a meta-evaluation methodology, where a pair of snippets that contain a *single* edit is given, and we compare the (reference-less) models according to the accuracy of the binary prediction (i.e., which of the snippets is a revision). By focusing on comparing 'single edit' differences, we can obtain transparent and interpretable measures for each type of edit (e.g., which edit type is more challenging to revise than other types). This is expected to enable us to investigate more effective evaluators (evaluation models) in the future. In fact, recent studies have demonstrated that such rubric-based interpretable evaluation correlates better with human judgments than single overall scoring techniques (Kasai et al., 2021a,b; Zhong et al., 2022). An overview of the proposed IRC is shown in Figure 2. The design philosophy of IRC is to provide users (e.g., model developers or writers) with analytical reports based on multidimensional evaluations to facilitate their understanding of the models, with the goal of moving away from chasing the highest overall number.

5.3 Experiment

In this subsection, we demonstrate how well existing large-scale pre-trained language models perform under the proposed IRC framework as (reference-less) models.

5.3.1 Data split

We divided TETRA into a training set (75%; 48 papers) and a test set (25%; 16 papers) to avoid paper overlap, and we converted the test data into pairs of snippets containing a single edit for IRC framework. Here, when multiple edit types were assigned, each edit type was extracted independently as a single edit snippet pair. When creating a pair of snippets,

	Grammaticality	**Fluency**	**Clarity**	**Style**	**Readability**	**Redudancy**	**Consistency**
BERT	**0.82**	**0.84**	**0.85**	**0.83**	**0.85**	0.79	**0.90**
GPT-4 zero-shot	0.42	0.57	0.55	0.59	0.46	0.47	0.58
+ explicit prompt	0.65	**0.79**	**0.67**	0.57	**0.71**	**0.92**	**0.62**
GPT-4 few-shot	0.47	0.48	0.56	0.51	0.44	0.45	0.40
+ explicit prompt	0.43	0.49	0.56	0.56	0.57	0.80	0.56

Table 6: Meta-evaluation result (Accuracy).

we extracted the entire paragraph as the context. In total, we extracted 1,368 snippet pairs for IRC meta-evaluation.

5.3.2 Evaluators

In this experiment, we compared BERT (Devlin et al., 2019) as fine-tuning and GPT-4 (OpenAI, 2023) as zero/few-shot settings to classify the original and single edit revision snippets.

BERT We converted the training set into a balanced positive/negative example by randomly swapping the order of snippet pairs in one-half of the training set. Specifically, we implemented this evaluator as a classification problem for the [CLS] tokens, using as input a sequence of tokens connecting the original and the single-edited revision documents with the [SEP] tokens. We used the PyTorch implementation for these Transformer models (Wolf et al., 2020). The hyperparameters used to train the model are shown in Appendix D

GPT-4 We build the model using the GPT-4 API (2024-02-15-preview) provided by OpenAI [7]. Two settings, zero-shot and few-shot (2-shot by following (Coyne et al., 2023)), were prepared to evaluate the performance with and without examples[8]. Furthermore, we created prompts focusing on text revision evaluation criteria (**explicit prompt**) to investigate the impact of prompts on evaluation performance, comparing them with the base prompt. Detailed information on each prompt is provided in Appendix E.

5.3.3 Results

As can be seen, the proposed IRC framework enabled us to evaluate the accuracy of each metric in terms of each aspect (i.e., edit type) while analyzing their strengths and weaknesses (Table 6). We also observe a significant disparity between fine-tuning and zero/few-shot results, highlighting the crucial role of fine-tuning in achieving automatic evaluation of text revision. Contrary to expectations, the LLM-based evaluator performed better in zero-shot compared to few-shot scenarios. One potential explanation is that presenting only a few cases might not only be insufficient but also noisy, especially in tasks involving diverse evaluation aspects and reasonable editing methods, such as text revision. On the other hand, compared to the base prompt, performance was significantly improved for many revision types when using explicit prompts. In particular, for redundancy, the GPT-4 evaluator with explicit prompt outperformed the finetuning model. This suggests the potential to realize an automatic evaluation model for high-performance text revision even for zero-shot by advancing prompt engineering in the future.

6 Analysis

The experimental results discussed in §5.3 demonstrated that the supervised metric can discriminate the original and revision snippets with reasonably high accuracy. However, the following question should be considered. *Is the high accuracy derived from actually detecting the quality improvement provided by the revision or annotation artifacts (spurious correlation) by commonly used words and phrases by expert annotators?*

To investigate this question, we evaluated the performance of *the same* supervised metric (BERT) used in §5.3 by applying corruption methods to TETRA in order to artificially degrade the quality of the source documents. If the same supervised metric is fine-tuned on the source and the (improved) revision can still select the original document over the degraded document, we can conclude that the metric actually distinguishes the *quality* of the document rather than spurious features.

6.1 Corruption Methods

Automatic Error Generation (AEG) Injecting grammatical errors as data augmentation has been studied actively to improve GEC. In this study, we

[7] https://github.com/openai/openai-python

[8] The example used for the few-shot was sampled from the train split.

used a back-translation model, which is the most commonly used model in GEC among AEG methods (Xie et al., 2018; Kiyono et al., 2019; Koyama et al., 2021), to deteriorate the original documents in terms of *grammaticality* and *fluency*. Here, a reverse model that generates an ungrammatical sentence from a given grammatical sentence was trained in the back-translation model. To construct the reverse model, we followed the general settings identified in previous studies (Kiyono et al., 2019; Koyama et al., 2021). The details of the experimental settings for the AEG model are described in Appendix F.

Sentence Shuffling As shown in Figure 1, the document revision process involves reordering sentences to improve the *flow* and *consistency* of argumentation. In this analytical experiment, after applying the AEG model, we further shuffled sentences with the same ratio as the *consistency* edit type (5% of the documents; refer to Table 3) to degrade the document relative to the sentence order.

6.2 Results

The binary classification accuracy obtained by BERT on the original vs. (degrading) corruption scenario was 0.96. We found that BERT can successfully select the original document over the degraded document. It should be noted that this is a simulation experiment with artificial errors and there are deviations from a realistic setting, but it suggests that the supervised baseline has the potential to learn to discriminate documents relative to quality rather than spurious features in the experts' annotations.

7 Conclusion

We have proposed the new document revision corpus and highlighted its uniqueness of it compared with existing corpora. As a case study using this corpus, we have explored reference-less and interpretable meta-evaluation methods and also demonstrated that a fine-tuned pre-trained language model can discriminate the quality of documents, which indicates the feasibility of automated document revision evaluation.

Limitations

The first limitation of this study is the scalability of the annotation. TETRA consists of *documents* revised by experts and is therefore expensive to scale up in its nature. This limitation could be mitigated by the choice of source data, i.e., there is room to replace experts with crowd workers by selecting source data that do not require expertise (e.g., general essays). We also reiterate that this work does not aim at proposing specific revision systems and evaluation models for automated document revision. Instead, we present a meta-evaluation scheme as a first step to develop such models and metrics with more transparency.

Ethics Statement

For developing a new document-level revision corpus, TETRA, we paid market rates to the professional editors for their annotations. With regard to the checklist items regarding the use and distribution of artifacts, none of the concerns apply to the dataset created in this study, as it was annotated based on the ACL Anthology materials. [9]

References

Linda Allal, Lucile Chanquoy, and Pierre Largy. 2004. *Revision Cognitive and Instructional Processes.*, volume 8. Springer.

Larry Beason. 1993. Feedback and revision in writing across the curriculum classes. *Research in the Teaching of English*, pages 395–422.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293.

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

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

M. Buchman, R. Moore, L. Stern, and B. Feist. 2000. *Power Writing: Writing with Purpose*. No. 4. Pearson Education Canada.

Jill Burstein, Daniel Marcu, and Kevin Knight. 2003. Finding the write stuff: Automatic identification of discourse structure in student essays. *IEEE Intelligent Systems*, 18(1):32–39.

[9] https://aclanthology.org/faq/copyright/

Aoife Cahill, Nitin Madnani, Joel Tetreault, and Diane Napolitano. 2013. Robust systems for preposition error correction using Wikipedia revisions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 507–517.

Hang Cao, Zhiquan Cao, Chi Hu, Baoyu Hou, Tong Xiao, and Jingbo Zhu. 2023. Improving autoregressive grammatical error correction with non-autoregressive models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12014–12027, Toronto, Canada. Association for Computational Linguistics.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30.

Shamil Chollampatt and Hwee Tou Ng. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5755–5762.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 124–129, New Orleans, Louisiana. Association for Computational Linguistics.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

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 (NAACL 2012)*, pages 568–572.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics.

Mike D'Arcy, Alexis Ross, Erin Bransom, Bailey Kuehl, Jonathan Bragg, Tom Hope, and Doug Downey. 2023. Aries: A corpus of scientific paper edits made in response to peer reviews.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Semire Dikli. 2006. An Overview of Automated Scoring of Essays. *Journal of Technology, Learning, and Assessment*, 5(1).

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590.

Lester Faigley and Stephen Witte. 1981. Analyzing revision. *College composition and communication*, 32(4):400–414.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACL-HLT 2015)*, pages 578–587.

Rachele De Felice and Stephen G. Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *COLING*, pages 169–176.

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

Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College composition and communication*, 32(4):365–387.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study. *arXiv preprint arXiv:1807.01270*.

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

Takumi 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.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise neural machine translation evaluation. 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 805–814. Association for Computational Linguistics.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering*, 12(2):115–129.

Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using Google Web 1T 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249.

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

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A Nested Attention Neural Hybrid Model for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 753–762.

Chao Jiang, Wei Xu, and Samuel Stevens. 2022. arXivEdits: Understanding the human revision process in scientific writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9420–9435, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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 (ACL 2020)*, pages 4248–4254.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander R. Fabbri, Yejin Choi, and Noah A. Smith. 2021a. Bidimensional leaderboards: Generate and evaluate language hand in hand. *arXiv* `https://arxiv.org/abs/2112.04139`.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2021b. Transparent human evaluation for image captioning. *arXiv* `https://arxiv.org/abs/2111.08940`.

Omid Kashefi, Tazin Afrin, Meghan Dale, Christopher Olshefski, Amanda Godley, Diane Litman, and Rebecca Hwa. 2022. Argrewrite v. 2: an annotated argumentative revisions corpus. *Language Resources and Evaluation*, pages 1–35.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

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 2019)*, pages 1236–1242.

Charles W. Kneupper. 1978. Teaching argument: An introduction to the toulmin model. *College Composition and Communication*, 29(3):237–241.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Aomi Koyama, Kengo Hotate, Masahiro Kaneko, and Mamoru Komachi. 2021. Comparison of grammatical error correction using back-translation models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 126–135.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 7(1):1–170.

John Lee and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182.

John Lee and Jonathan Webster. 2012. A corpus of textual revisions in second language writing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 248–252, Jeju Island, Korea. Association for Computational Linguistics.

Yinghao Li, Xuebo Liu, Shuo Wang, Peiyuan Gong, Derek F. Wong, Yang Gao, Heyan 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, Toronto, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Liu, Alexander R. Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization.

Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, and Kentaro Inui. 2019. Cross-corpora evaluation and analysis of grammatical error correction models — is single-corpus evaluation enough? 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 1309–1314, Minneapolis, Minnesota. Association for Computational Linguistics.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. 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 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. Shared task on feedback comment generation for language learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A Feedback-Augmented Method for Detecting Errors in the Writing of Learners of English. In *COLING-ACL*, pages 241–248.

Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764.

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*, pages 551–566.

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

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Referenceless evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 229–234.

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 18th Conference on Computational Natural Language Learning (CoNLL 2014): Shared Task*, pages 1–14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013): Shared Task*, pages 1–12.

OpenAI. 2023. Gpt-4 technical report.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

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 and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Anthony Seow. 2002. *The Writing Process and Process Writing*, page 315–320. Cambridge University Press.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and Aspect Error Correction for ESL Learners Using Global Context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 198–202.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872.

Kristina Toutanova and Robert Moore. 2002. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 31 (NIPS 2017)*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Ziang Xie, Guillaume Genthial, Andrew Y. Ng, and Dan Jurafsky. 2018. Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *NAACL*, pages 619–628.

Lingyu Yang, Hongjia Li, Lei Li, Chengyin Xu, Shutao Xia, and Chun Yuan. 2023. LET: Leveraging error type information for grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5986–5998, Toronto, Canada. Association for Computational Linguistics.

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

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Fan Zhang, Homa B. Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578. Association for Computational Linguistics.

Fan Zhang, Rebecca Hwa, Diane Litman, and Homa B. Hashemi. 2016. ArgRewrite: A web-based revision assistant for argumentative writings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 37–41, San Diego, California. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2014. Sentence-level rewriting detection. In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–154, Baltimore, Maryland. Association for Computational Linguistics.

Fan Zhang and Diane Litman. 2015. Annotation and classification of argumentative writing revisions. In

Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 133–143, Denver, Colorado. Association for Computational Linguistics.

Yue Zhang, Leyang Cui, Enbo Zhao, Wei Bi, and Shuming Shi. 2023. RobustGEC: Robust grammatical error correction against subtle context perturbation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16780–16793, Singapore. Association for Computational Linguistics.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation.

A Recruitment procedure for annotators

We recruited professional editors who are native speakers of English and have domain expertise in academic writing, directly via Upwork (`https://www.upwork.com/`), a freelance marketplace, through interviews and screening tests to ensure the quality of the annotators. We paid market rates to them. Instead of using the services of an English proofreading company, which tends to be uncontrollable in terms of annotator quality, we directly hired annotators and provided them with feedback to control the annotation quality, which contributed to further improving the dataset's quality. We will extend the description of this annotation process in the camera ready.

B Instructions for annotators

The full text of the instructions to the annotators is reported below.

Summary You will be proofreading and editing the abstracts and the introduction sections of scientific papers published at NLP (Natural Language Processing) conferences and workshops. Please make edits to improve the quality of the papers, along with your comments mentioning what aspect of the paper the edit is intended to improve, without changing the meaning of the content (information contained in the paper).

About the papers

- These papers are randomly chosen from a pool of papers published at recent NLP conferences and workshops.
- These papers are written by a diverse set of authors, including native and non-native speakers of English at various stages of their careers (students, researchers, faculty members, etc.).
- These papers went through peer reviews and were accepted at conferences and workshops

Edits

- Make edits to the papers in order to improve their quality without changing the information contained in the papers. For each edit, mention what aspect of the paper the edit is intended to improve. These aspects include, but are not limited to: Mechanics, punctuation, grammar, spelling, word order, word usage, organization, development, cohesiveness, coherence, clarity, content, consistency, voice. Feel free to use your own tags/words to describe the purpose of your edit
- Refrain from making single edits that improve more than one aspect of the paper at the same time. Make two or more separate, overlapping edits in the same place if you need to improve multiple aspects.
- Feel free to be creative and make changes that span over multiple sentences or ones that rearrange sentences or even paragraphs if necessary. You are encouraged to rewrite the sentences and paragraphs if local edits aren't enough to improve the quality.
- Since these papers are already peer-reviewed, we expect fewer low-level edits related to punctuation, spelling, and grammar, although make sure to correct such errors if you do encounter them.
- Focus instead on types of edits that improve higher-level aspects of the paper (such as organization, development, cohesiveness, coherence, clarity, content, voice, etc.)

C Example of XML annotation

See Table 7.

D Hyper-parameters settings

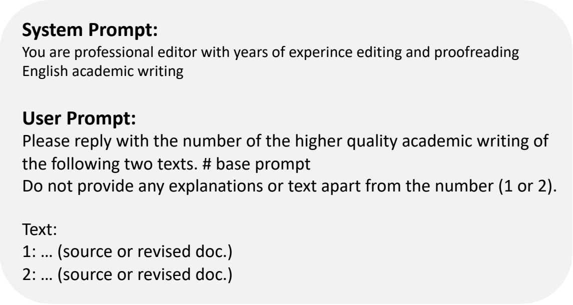
See Table 8.

E Prompts in the GPT-4 evaluators

The prompt used for GPT-4 evaluator is illustrated in Table 3. For prompts focused on evaluation criteria, the following sentence was replaced with base prompt.

```
<doc id="Pxx-xxxx" editor="A" format="Conference" position="Non-student" region="Native">
<abstract>
<text>In this paper, (...) extracted sense inventory. The</text>
<edit type="conciseness" crr="induction and disambiguation steps" comments="conciseness - just
    tightening it up a little bit.">induction step and the disambiguation step</edit>
<text>are based on the same principle: (...) topical dimensions</text>
<edit type="readability" crr=". In" comments="readability - this sentence is getting a bit long, so
    splitting it in two here.">; in</edit>
<text>a similar vein, ...</text>
...
</abstract>
<introduction>
<text>Word sense induction (...)</text>
<text>\n\n Word sense disambiguation (...)</text>
<edit type="punctuation" crr="" comments="punctuation - comma is not appropriate.">,</edit>
...
</introduction>
```

Table 7: Example of XML annotation. For brevity, we omitted a part of the text with "...".



System Prompt:
You are professional editor with years of experince editing and proofreading English academic writing

User Prompt:
Please reply with the number of the higher quality academic writing of the following two texts. # base prompt
Do not provide any explanations or text apart from the number (1 or 2).

Text:
1: ... (source or revised doc.)
2: ... (source or revised doc.)

Figure 3: Example of prompt.

Configurations	Values
Model Architecture	bert-base-uncased
Optimizer	Adam (Kingma and Ba, 2015)
Learning Rate	2e-5
Number of Epochs	10
Batch Size	32

Table 8: Hyper-parameters settings

- Grammaticality: "Please reply with a more grammatical text number."
- Fluency: "Please reply with a more fluent text number."
- Clarity: "Please reply with the number of the text whose meaning is clearer."
- Style: "Please reply with the number of the higher quality academic writing of the following two texts. Please focus your evaluation on the adaptation to an academic writing style in particular."
- Readability: "Please reply with a more readable text number."
- Redundancy: "Please reply with a text number that is less redundant."
- Consistency: "Please reply with more consistent text."

F Experimental settings for AEG

We adopted the "Transformer (big)" settings (Vaswani et al., 2017) using the implementation in the fairseq toolkit (Ott et al., 2019) as a GEC model. In addition, we used the BEA-2019 workshop official dataset (Bryant et al., 2019) as the training and validation data. For preprocessing, we tokenized the training data using the spaCy tokenizer[10]. Then, we removed sentence pairs where both sentences where identical or both longer than 80 tokens. Finally, we acquired subwords from the target sentence via the byte-pair-encoding (BPE) (Sennrich et al., 2016) algorithm. We used the subword-nmt implementation[11] and then applied BPE to split both source and target texts. The number of merge operations was set to 8,000.

[10] https://spacy.io/

[11] https://github.com/rsennrich/subword-nmt