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

Anonymous ACL submission

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. 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, i.e., **Tex**t **R**evision of **A**CL papers (TETRA), in which professional editors have revised academic papers sampled from the ACL anthology that contain trivial grammatical errors. This corpus enables us to focus on document-level and paragraph-level edits, e.g., edits related to coherence and consistency. In addition, 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 an important process in essay and argumentative writing. According to previous studies on argumentative writing (Flower and Hayes, 1981; Beason, 1993; Buchman et al., 2000; Seow, 2002; Allal et al., 2004), a typical writing process comprises three main stages. *Revising* is the initial editing step to plan and build the overall structure of the document at a high level, *editing* focuses on sentence-level or phrase-level expressions, and *proofreading* is performed to assess the details, e.g., spelling and grammatical errors (Figure 1, left). Although the order of the steps is not strictly determined, the typical writing process starts from a broad and high level perspective and then narrows down the scope of edits.

In contrast to the typical human writing process, automated grammatical error correction (GEC) research in the natural language processing (NLP) field 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). Then, the research community expanded its scope to edits at the phrase and sentence levels while considering fluency (Sakaguchi et al., 2016; Napoles et al., 2017) (Figure 1, right). However, much less work has been done on the *document-level* revisions due to two major challenges. First, document revisions have a broader scope (e.g., coherence and flow) than conventional GEC and fluency correction; thus, there are few publicly available corpora with such annotations by experts (e.g., professional editors). Second, it is challenging to evaluate the quality of revisions based on a limited number of references because there are many ways to revise a single document. This implies that *reference-less* evaluation metrics (Napoles et al., 2016b; Choshen and Abend, 2018; Islam and Magnani, 2021) are suitable to assess automated document revision models.

We consider these challenges associated with automated document revision, propose a new corpus, and exploring possibilities for transparent evaluation methods that are independent of gold standards or references. Our corpus, i.e., **Tex**t **R**evision of **A**CL papers (TETRA),[1] comprises document-level revisions of articles published at ACL-related venues. This corpus was designed based on an annotation scheme that can handle edit types beyond sentences, e.g., argument flow, in addition to

[1] https://github.com/anonymous

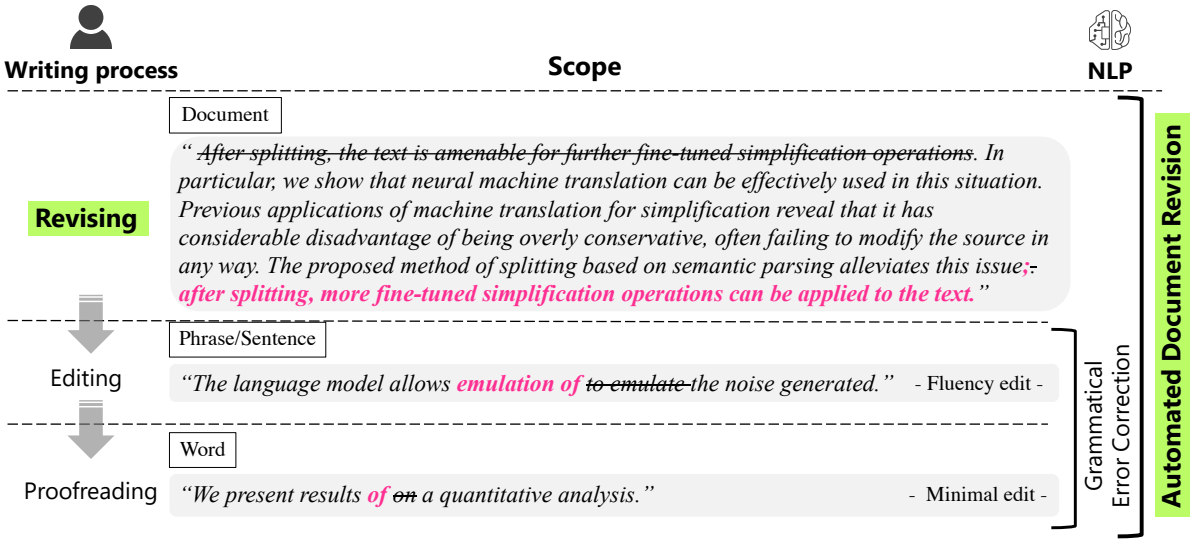

Figure 1: Overview of the scope for automated document revision. Each example is taken from TETRA corpus. We focus document revision process which has been overlooked by grammatical error correction (GEC). Automated document revision extends the scope of GEC.

conventional word-level and phrase-level edits.

In this paper, we demonstrate that TETRA has advantages over existing corpora for document revision (Lee and Webster, 2012; Zhang et al., 2017; Kashefi et al., 2022). We also propose a simple meta-evaluation method, i.e., **i**nstance-based **r**evision **c**lassification (IRC), that measures and compares the performance of evaluation metric candidates based on their accuracy in terms of classifying which of a given pair of snippets has been revised. The accuracy for IRC is calculated for type of edit, which provides transparent and interpretable analyses to design more effective evaluation metrics in future. Note that our contribution is not to propose a specific model or metric for automated document revision but rather present a meta-evaluation scheme to help measure the ongoing improvements of such models and metrics.

With the proposed TETRA corpus and IRC, we conducted experiments to determine whether pretrained language models can function as an effective baseline metric to discriminate between original and revised snippets. Here, we compared BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) as baseline methods with and without fine tuning. The results demonstrate that the supervised method can select better snippets with an accuracy of 0.85 – 0.96, which indicates the feasibility of evaluation for automated document revision.

2 Background

The GEC field, 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] Initial GEC systems focused on only a small number of closed-class error types, e.g., 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 then 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 work resulted in 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 make the result 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* edits (Sakaguchi et al., 2016). With this expansion, the community has proposed new benchmark datasets (Napoles et al., 2017; Bryant et al., 2019; Napoles et al., 2019; Flachs et al., 2020) and evaluation metrics (Dahlmeier and Ng, 2012;

[2] In this paper, we focus on GEC literature after the 2000's, at which point statistical methods began to be applied widely. For the full history of GEC in the 80's and 90's, e.g., rule-based approaches, please refer to Leacock et al. (2014).

Felice and Briscoe, 2015; Napoles et al., 2015; Bryant et al., 2017; Napoles et al., 2019; Gotou et al., 2020) 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). More recently, Yuan and Bryant (2021) extended DNN models by taking a longer context (e.g., previous sentences) and demonstrated improvements in terms of sentence-level error correction (e.g., correcting verb tense).

In contrast to the rapid progress of grammar and fluency correction, few studies have investigated revisions for *document-level argumentative writing*, which requires more human effort to create corpora or datasets. Lee and Webster (2012) performed 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 prepared 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 because the corpus comprises writing from language learners, and the vast majority of errors were simple grammar and fluency errors. This provides an important lesson, i.e., a corpus for document-level revision should be based on documents in which grammatical and fluency edits have already been addressed to some degree. In addition, due to copyright limitations, this corpus is not publicly available. However, we believe that the data source for a document-level corpus should be accessible under an open license to promote long-term community-based open research.

Another line of work (Zhang and Litman, 2014, 2015; Zhang et al., 2016, 2017; Kashefi et al., 2022) has developed the ArgRewrite corpus, which is a collection of 86 argumentative essays, each of which comprises three drafts (i.e., two cycles of revisions) with edit labels. In the ArgRewrite corpus (both v1 and v2), approximately one-half of all edits are annotated as surface-level corrections (i.e., conventional GEC or fluency edits), and the remaining edits are annotated as content-level document revisions. The ArgRewrite corpus has advantages over Lee and Webster (2012) in terms of the amount of document-level revisions; however, all of the essays in the ArgRewrite corpus were written on the same topic. The topic of the first version (Zhang et al., 2017) is about arguing *whether the proliferation of electronic enriches or hinders the development of interpersonal relationships*, and the topic of the second version (Kashefi et al., 2022) is about arguing *support or against self-driving cars*. Relative to developing and evaluating automated document revision models, this limitation in terms of topic diversity can cause overfitting (Mita et al., 2019).

3 Automated Document Revision

Given a source document d that consists of paragraphs, a potentially automated editor f revises 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. We discuss our meta-evaluation framework in §5.

4 The TETRA Corpus

4.1 Data Source

In this study, we used the ACL anthology[3] as the data source for the proposed TETRA for the following reasons. First, we focus on *document-level* revision rather than sentence-level revision; thus,

[3] https://aclanthology.org

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

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 2: Examples of revision. Each edit type is highlighted respectively.

we selected documents that have as few grammatical errors (i.e., the conventional scope of GEC) as possible. The ACL anthology comprises generally well-written peer-reviewed papers on NLP. Second, the ACL anthology contains a diverse range of papers in terms of authors and venues, e.g., conferences vs. workshops, students vs. nonstudents, native vs. nonnative English speakers, as demonstrated by Bergsma et al. (2012). Finally, the license and copyright of the ACL anthology are more flexible than existing datasets for similar purposes (Lee and Webster, 2012). Note that a less restricted and widely accessible corpus enables us to advance research on automated document revision.

We selected the source documents from the ACL anthology as follows. First, we created eight (=2^3) groups 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. nonnative English speaking country, and (3) whether the first author was a student (at the time the paper was published). We randomly sampled the papers until we obtained eight unique papers for each group (i.e., 64 papers in total). For each paper, we extracted the title, abstract, and introduction as the source document (d) of the proposed TETRA corpus.

4.2 Annotation Scheme

The scope and granularity of edit types also has a wide variety in previous studies, and there is no standard set of labels. Thus, we define edit type categories (Table 1) based on previous literature on argumentative and discourse writing (Kneupper, 1978; Faigley and Witte, 1981; Burstein et al., 2003; Zhang et al., 2017). Table 2 shows 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)

	Lee and Webster (2012)	Zhang et al. (2017)	Kashefi et al. (2022)	Ours (TETRA)
# docs	3760	60	86	64
# references	1	1	1	3
% beyondGECs	3.2	49.4	52.6	56.9
Drafted by	ESL	ESL/Native	ESL/Native	ESL/Native
Revised by	Author	Author	Author	Experts
Feedback by	Non-experts	Experts	Experts	Experts
Topic diversity	✓			✓
Public availability		✓	✓	✓

Table 3: Characteristics of TETRA corpus compared to existing document revision corpus. % 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 by whose feedback (*feedback by*). 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.

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, e.g., edit types, edit rationale, comments, and other meta-information. An example of our XML annotation is shown in the Appendix (Table 6).

4.3 Annotators

We recruited three native English speaking professional editors with years of experience editing and proofreading English academic writing. These editors independently revised all 64 documents on the Google Docs platform, and they added an edit rationale whenever appropriate. The revised documents were converted to XML format by the first two authors.[4] Details on how to recruit annotators and instructions for them are provided in Appendix A and B, respectively.

4.4 Statistical Analysis

The right-most column in Table 1 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 3). 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 in the Appendix (Table 7).

[4]While converting, we made minor corrections and remapping edit types only as required.

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

Table 4: Two levels of inter-annotator agreement: agreement on *detection* and *correction*.

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 4 shows the results.

The result demonstrate 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 Proposed Meta-evaluation Framework

In addition to creating a corpus for automated document revision, it is essential to establish evaluation

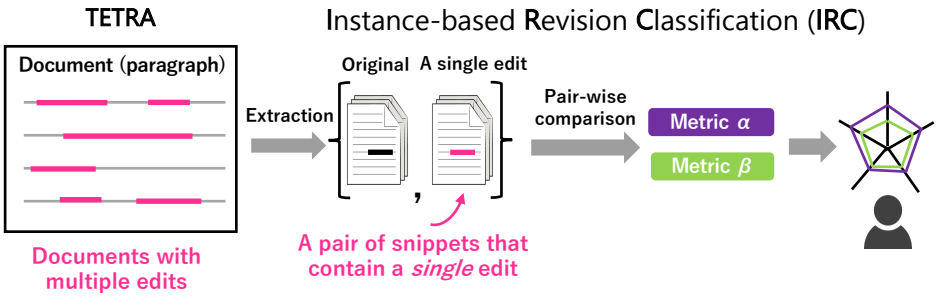

Figure 2: Overview of meta-evaluation framework. We introduce document revision corpus (TETRA) and propose instance-based revision classification (IRC) to measure (meta-evaluate) the quality improvement of documents.

metrics that can measure a document's quality improvement (and possibly deterioration) relative to the applied revisions. 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 (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 addition, 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 1). 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 the improvement or the degree of improvement. To address these concerns, we propose **I**nstance-based **r**evision **c**lassification (IRC), where a pair of snippets that contain a *single* edit is given, and we compare the (reference-less) metrics according to the accuracy of the binary prediction (i.e., which of the snippets is 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 evaluation metrics in 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). An overview of the proposed IRC is shown in Figure 2.

6 Experiment

In this section, we demonstrate how well existing large-scale pretrained language models perform under the proposed IRC framework as baseline (reference-less) metrics.

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

6.2 Baseline metrics

In this experiment, we compared BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) as supervised and unsupervised settings to classify the original and single edit revision snippets. We used the PyTorch implementation for these Transformer models (Wolf et al., 2020).

BERT We adopted BERT (with fine tuning) as a supervised evaluation metric. Here, 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 concatenated the pair with a special [SEP] token. The hyperparameters used to train the model are shown in Appendix E (Table 8).

GPT-2 We adopted GPT-2 as an unsupervised baseline metric. Here, we compared the per-token

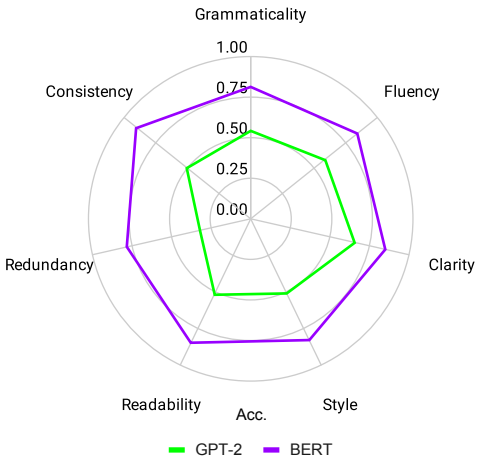

Figure 3: Meta-evaluation result (Accuracy).

perplexities of the two inputs (i.e., the original and the single-edited revision), and we used them to perform binary prediction. Technically, we expected that the revised documents would show lower per-token perplexity (and vice versa).

6.3 Results

The overall results are shown in Figure 3. 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. We also found that the supervised metric (i.e., BERT) could perform classification at 0.79 – 0.90 accuracy, which indicates that this supervised metric based on pretrained neural language models is an effective baseline metric to discriminate between the original and revised snippets even when the difference is subtle. However, we found that the unsupervised baseline metric (i.e., GPT-2) performed slightly better than the chance-level only on grammaticality, fluency, and clarity but not on consistency, readability, and style.

7 Analysis

7.1 Is IRC framework reliable?

The experimental results discussed in §6 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 §6 by applying corruption methods to TETRA in order to artificially degrade the quality of the source documents. If the same supervised metric 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.

7.1.1 Corruption Methods

Automatic Error Generation (AEG) Injecting grammatical errors as data augmentation has been studied actively to improve GEC. In this study, we used a back-translation model, which is the most commonly model used 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 the 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 1) to degrade the document relative to the sentence order.

7.1.2 Results

The binary classification accuracy obtained by the supervised metric (BERT) used in the experiment (§6) 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.

Outputs	ERRANT	GLEU
original (no editing)	0.0 (0.0)	70.6 (1.5)
human performance	24.5 (5.7)	71.4 (1.0)

Table 5: Evaluation results with GEC's metrics.Values in parentheses indicate standard deviations.

7.2 Do existing GEC metrics not work?

In §5, we hypothesized that common reference-based GEC metrics cannot evaluate document revisions accurately. To verify this hypothesis, we evaluated the gold revisions of human experts in TETRA using existing GEC metrics, and we analyzed whether they actually work for the revision of documents.

7.2.1 Examined metrics

We used ERRANT (Bryant et al., 2017) and GLEU (Napoles et al., 2015, 2016a), which are widely used in GEC, as the examined metrics.

ERRANT ERRANT is an improved version of the previously standard Max Match (M^2) Scorer metric (Dahlmeier and Ng, 2012). Similar to the M^2 Scorer, ERRANT is performed based on the Max Match method, which identifies the maximum match using an edit lattice when matching edits between systems and references; however, the method of edit extraction in ERRANT differs from that used in the M^2 Scorer.

GLEU GLEU is a variant of BLEU (Papineni et al., 2002), which is *de facto* evaluation metric in machine translation, for GEC. GLEU is computed by subtracting the number of n-grams that appear in the input but not in the reference from the number of n-grams that match in the system output and reference. This metric is more highly correlated with human judgment than the M^2 Scorer (Napoles et al., 2016b).

7.2.2 Results

Table 5 shows the evaluation results when the original documents (original) and gold revisions by experts (human performance) were regarded as the system outputs. Here, three gold revisions by the human experts were assigned to TETRA; thus, the values represent the average of the three. The evaluation results obtained with ERRANT demonstrate that even human performance has a low value of 24.5 points (out of 100), which implies that it has issues evaluating document revisions. ERRANT evaluates systems based on the extent to which the edit span suggested by systems matches the gold edit span included in the given references. However, in document revisions that require cross-sentence editing or more dynamic editing, ERRANT may have difficulty extracting accurate edit spans and matching them with the references.

In contrast, GLEU may appear to work as an evaluation metric because it gives higher scores to human performance. However, GLEU also has issues because its evaluation score for original documents, i.e., outputs without editing, is comparable to that of the human experts. The GLEU score was computed based on the n-gram agreement ratio in the three sentences (documents in our case), i.e., the input, the system output, and the reference. In document revision, a task with low agreement rates (§4.4), GLEU, which performs document-by-document matching, tends to overestimate the unedited output.

8 Limitations

The first limitation of this study is its 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 models and evaluation metrics for automated document revision. Instead, we present a meta-evaluation scheme as a first step to develop such models and metrics with more transparency.

9 Conclusion

In this paper, we have proposed a new automated document revision task, and we have also proposed the TETRA corpus and the IRC meta-evaluation method, which facilitates interpretable analysis to support designing more effective evaluation metrics without reference. Our experimental results demonstrate that a fine-tuned pretrained language model can discriminate the quality of documents even when there is only a single edit, which indicates the feasibility of automated document revision evaluation. We hope that the proposed TETRA and IRC will encourage the community to further study automated document revision models and metrics beyond sentence-level error corrections.

Ethics Statement

For developing a new document-level revision corpus, TETRA, we paid market rates to the professional editors for their annotations.

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.

Shane Bergsma, Matt Post, and David Yarowsky. 2012. Stylometric analysis of scientific articles. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 327–337, Montréal, Canada. Association for Computational Linguistics.

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.

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.

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.

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.

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.

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

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*.

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.

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.

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.

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.

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, Matt Post, and Joel Tetreault. 2016a. GLEU Without Tuning. *arXiv preprint arXiv:1605.02592*.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016b. There's no comparison: Reference-less 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.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

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 gram-

matical 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.

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.

Zheng Yuan and Christopher Bryant. 2021. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.

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.

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 6.

D Aspect distribution

See Table 7.

E Hyper-parameters settings

See Table 8.

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[5]. 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

[5] https://spacy.io/

```
<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 6: Example of XML annotation. For brevity, we omitted a part of the text with "...".

	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 7: Distributions of revision aspects by writer's attributes.

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

(BPE) (Sennrich et al., 2016) algorithm. We used the `subword-nmt` implementation[6] and then applied BPE to split both source and target texts. The number of merge operations was set to 8,000.

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