Diamonds in the Rough: Generating Fluent Sentences from Early-Stage Drafts for Academic Writing Assistance

Takumi Ito[*,1,2] , **Tatsuki Kuribayashi**[*,1,2], **Hayato Kobayashi**[3,4], **Ana Brassard**[4,1], **Masato Hagiwara**[5], **Jun Suzuki**[1,4], and **Kentaro Inui**[1,4]
[1]Tohoku University [2]Langsmith Inc. [3]Yahoo Japan Corporation [4]RIKEN [5]Octanove Labs LLC
{t-ito, kuribayashi, jun.suzuki, inui}@ecei.tohoku.ac.jp
hakobaya@yahoo-corp.jp, ana.brassard@riken.jp
masato@octanove.com

Abstract

The writing process consists of several stages such as drafting, revising, editing, and proofreading. Studies on writing assistance, such as grammatical error correction (GEC), have mainly focused on sentence *editing* and *proofreading*, where surface-level issues such as typographical, spelling, or grammatical errors should be corrected. We broaden this focus to include the earlier *revising* stage, where sentences require adjustment to the information included or major rewriting and propose *Sentence-level Revision (SentRev)* as a new writing assistance task. Well-performing systems in this task can help inexperienced authors by producing fluent, complete sentences given their rough, incomplete drafts. We build a new freely available crowdsourced evaluation dataset consisting of incomplete sentences authored by non-native writers paired with their final versions extracted from published academic papers for developing and evaluating SentRev models. We also establish baseline performance on SentRev using our newly built evaluation dataset.

1 Introduction

Academic writing can be a daunting task, even for experienced writers with a native or near-native command of English. Inexperienced, non-native speakers find themselves in an even more difficult situation—in addition to grammatical or spelling errors, their sentences may lack fluidity, have an awkward style, contain collocation errors, or have missing words where they could not remember or did not know the appropriate expressions. Such authors, especially students with insufficient academic experience, may often have difficulty putting their ideas and findings into words, even if the ideas are sound and contribute to the research community. Improving writing quality is thus a concern for both individual researchers and the academic community.

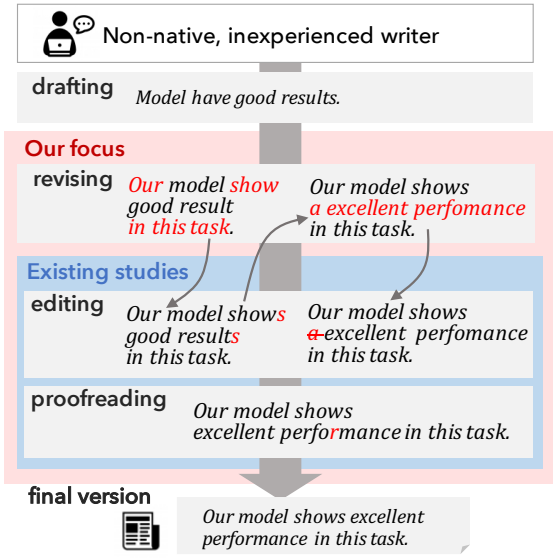

Figure 1: Overview of the estimated process of writing a sentence *Our model shows excellent performance in this task.*. Writing activity consists of four stages: (i) drafting, (ii) revising, (iii) editing, and (iv) proofreading.

Writing assistance technologies have been extensively studied in the natural language processing (NLP) field (Brill and Moore, 2000; Ng et al., 2014; Grangier and Auli, 2018). We focus on helping inexperienced authors in writing fluent grammatical sentences.

Models developed for academic writing assistance using existing datasets can serve as a support system during the final stages by editing a nearly finished version of the draft. For example, Daudaravicius (2015) collects scientific papers before and after professional editing from publishing companies, and Dale and Kilgarriff (2011) extract already published papers that still contain errors and correct the errors to obtain target fragments of text.

[*] The authors contributed equally

Process-writing pedagogy, however, asserts that writing comprises several processes (Susser, 1994; Seow, 2002; Buchman et al., 2000) as shown in Figure 1. This study takes on the challenge of automatic assistance in both the final checking process (*proofreading* and *editing*) and the earlier stages of writing (*revising*). In the revising stage, authors may drastically modify the wording and supplement some words, a highly demanding task for non-native or less experienced writers. Assistance in this stage has been less explored in NLP.

In this study, we design a new type of academic writing assistance task, Sentence-level Revision (SentRev), where a system receives an early draft of a sentence, and generates a revised, error-free, proofread version.

A critical issue in tackling this type of assistance task is that evaluation resources are scarce since early-stage draft sentences are not usually publicly available. To overcome this limitation, we release an evaluation dataset of pairs of draft sentences and their final versions, the *Set of Modified Incomplete TecHnical paper sentences* (SMITH), that we created using crowdsourcing techniques. Additionally, we evaluate the quality of our dataset and extensively analyze the characteristics of the obtained drafts. Finally, we train unsupervised models and report the baseline performance for our task on the SMITH evaluation dataset.

Our contribution is fourfold:

- We propose a new task—SentRev.
- We create an evaluation dataset, SMITH, for SentRev using a new crowdsourcing approach and release it.[1]
- We compare the characteristics of our dataset with major corpora and analyze the obtained draft sentences.
- We establish baseline scores for SentRev.

2 The Sentence-level Revision task

The proposed task, SentRev, is revising and editing incomplete draft sentences to create final versions. Examples of sentence-level revision are shown in Table 1.

A draft sentence, x, may have several types of problems. Surface-level problems such as typographical errors, spelling errors, or grammatical errors are a common occurrence. Wording problems, such as collocation errors or expressions being stylistically odd or inappropriate for the academic domain, are also typical of rough sentences written by non-native, inexperienced writers. The third type of error is *information gaps*. Information gaps are cases where the author likely could not find the appropriate wording for the idea he or she wanted to convey, such as a specific expression common in the academic domain or a technical term. In addition, a draft sentence may be missing sections without the author being aware of this. Solving the aforementioned problems in a draft sentence would elevate the draft sentence x to its final or nearly final version y with greatly improved correctness and fluency. Ideally, a single error-free and correctly filled-in final version should be generated while considering the context of the sentence. However, as a first step, an assistance system may output a set of *likely candidates* for the user to choose from or be inspired by, which would be realistic for a real-world application.

Draft	*However, the F1 score of KBP 2017 coupus* **<*>** *decreased by the sub event base rule.*
Reference	*However, subevent based constraints slightly reduced the F1 scores on KBP 2017 corpus.*
Draft	*But, there are some important difference to* **<*>** *our work unique.*
Reference	*However, there exist several key differences that make our work unique.*

Table 1: Examples of sentence-level revisions in our SMITH dataset. Our task is to transform the draft sentences into their corresponding reference sentences.

Our proposed task is, therefore, to generate likely final versions y from early-draft sentences x. For this purpose, we provide an evaluation dataset, SMITH, comprising pairs of drafts and their final versions (X, Y).

3 The SMITH dataset

3.1 Dataset creation

Process overview Although we cannot collect "drafts" X from published papers, we can easily collect the "final versions" Y. We also have access to non-native, inexperienced writers through crowdsourcing services. Our test set creation process combines these two factors (Figure 2). The

[1] https://github.com/taku-ito/INLG2019_SentRev

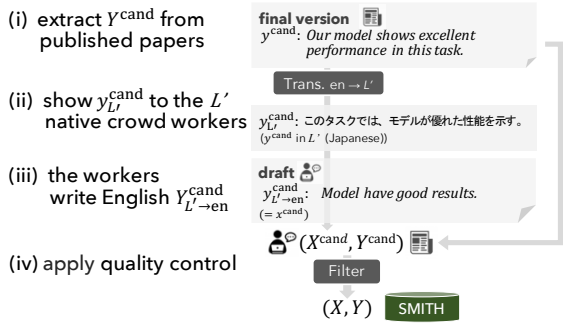

Figure 2: Overview of the crowdsourcing protocol for creating an evaluation dataset for the SentRev task.

protocol consists of the following four phases:

(i) Collecting a large number of sentences written by experts Y^{cand} from published papers.

(ii) Translating them into another language L', resulting in sentences $Y^{\text{cand}}_{L'}$.

(iii) Asking native speakers of L' to translate $Y^{\text{cand}}_{L'}$ back into English $Y^{\text{cand}}_{L'\to\text{en}}$ through crowdsourcing. Henceforth, we denote $Y^{\text{cand}}_{L'\to\text{en}}$ as X^{cand}.

(iv) Filtering the pairs of $(X^{\text{cand}}, Y^{\text{cand}})$ to ensure the quality of the dataset (X, Y).

This setting is analogous to the situation non-native writers face, as Cohen and Brooks-Carson (2001) report that non-native speakers tend to formulate in their native language and mentally translate to the target second language. We assume that most crowdworkers have never written an academic paper, and that the target users of SentRev-based systems also include this type of inexperienced writers.

To control the quality of the drafts, we first create many candidate pairs of drafts and reference sentences $(X^{\text{cand}}, Y^{\text{cand}})$ and then filter them to create the quality-controlled set (X, Y). The following subsections detail this process.

Collecting final version sentences We collected sentences Y^{cand} from the ACL Anthology Sentence Corpus (AASC).[2] We extracted the sentences that satisfied the following conditions from the AASC as Y^{cand}:

- accepted to ACL 2018,
- 70 to 120 characters long,
- does not include mathematical symbols, special tokens for citations, URLs, Greek letters, or other special symbols defined in AASC, and
- free of clear conversion mistakes when automatically extracted from PDFs.

Creating draft sentences We used Japanese as L'. First, we translated Y^{cand} into Japanese using Google Translate.[3] We denote the Japanese versions of Y^{cand} by $Y^{\text{cand}}_{\text{ja}}$. To guarantee the quality of $Y^{\text{cand}}_{\text{ja}}$, two authors of this paper, who were native speakers of Japanese, inspected all the sentences from $Y^{\text{cand}}_{\text{ja}}$ and removed those that at least one speaker judged to be incorrect translations.

Next, we asked each Japanese crowdworker to translate three sentences from $Y^{\text{cand}}_{\text{ja}}$ into English $Y^{\text{cand}}_{\text{ja}\to\text{en}}$ within 15 minutes. The appropriate time limit and rules were determined based on several trial tasks.

The workers were allowed to insert the special symbol <⋆> in places where they could not think of a good expression for that position in their answer $Y^{\text{cand}}_{\text{ja}\to\text{en}}$. This instruction revealed the information gaps that the authors of the drafts consciously left empty. An author may also be unaware that a draft sentence is missing sections. 306 workers participated in our crowdsourcing task.

Quality control We designed thorough filtering criteria and applied them to the workers because Yahoo! crowdsourcing, [4] a Japanese crowdsourcing service, does not provide filtering based on the worker's writing skills or abilities. We filtered workers depending on their writing activities. We scored each worker using the three answers they produced by using the criteria detailed in Table 2. We accepted work from workers with score 0 or higher as valid. The hyperparameters were determined with trial experiments. We used spaCy-CLD[5] for language detection.

In addition, to remove instances with a too large gap, we automatically filtered out the obtained $(x^{\text{cand}}, y^{\text{cand}}) \in (X^{\text{cand}}, Y^{\text{cand}})$ whose unigram overlap coefficient was considerably low:

$$\frac{|U(x^{\text{cand}}_{\text{checked}}) \cap U(y^{\text{cand}})|}{\min\{|U(x^{\text{cand}}_{\text{checked}})|, |U(y^{\text{cand}})|\}} < \alpha \ ,$$

where $U(\cdot)$ is the set of tokens excluding stopwords and special tokens (<⋆>). $x^{\text{cand}}_{\text{checked}}$ is the

[2] https://github.com/KMCS-NII/AASC

[3] https://translate.google.com/

[4] https://crowdsourcing.yahoo.co.jp/

[5] https://github.com/nickdavidhaynes/spacy-cld

Criteria	Judgment
Working time is too short (< 2 minutes)	Reject
All answers are too short (< 4 words)	Reject
No answer ends with "." or "?"	Reject
Contain identical answers	Reject
Some answers have Japanese words	Reject
No answer is recognized as English	Reject
Some answers are too short (< 4 words)	-2 points
Some answers use fewer than 4 kinds of words	-2 points
Too close to automatic translation (20 <= L.D. <= 30)	-0.5 points/ans
Too close to automatic translation (10 <= L.D. <= 20)	-1.5 points/ans
Too close to automatic translation (L.D. <= 10)	Reject
All answers end with "." or "?"	+1 points
Some answers have **<*>**	+1 points
All answers are recognized as English	+1 points

Table 2: Criteria for evaluating workers. L.D denotes the Levenshtein distance.

spell-checked version[6] of x^{cand}. α is set to 0.4, which was determined in trial experiments.

We collected 10,804 pairs of draft and their final versions, which cost us approximately US$4,200, including the trial rounds of crowdsourcing.

Unfortunately, works produced by unmotivated workers could have evaded the aforementioned filters and lowered the quality of our dataset. For example, workers could have bypassed the filter by simply repeating popular phrases in academic writing ("We apply we apply"). To estimate the frequency of such examples, we sampled 100 (x, y) pairs from (X, Y) and asked an NLP researcher (not an author of this paper) fluent in Japanese and English to check for examples where x was totally irrelevant to x_{ja}, which was shown to the crowdworkers when creating x. The expert observed no completely inappropriate examples, but noted a small number of clearly subpar translations. Therefore, 95% of sentence pairs were determined to be appropriate. This result shows that, overall, our method was suitable to create the dataset and confirms the quality of SMITH.

3.2 Statistics

Table 3 shows the statistics of our SMITH dataset and a comparison with major datasets for building a writing assistance system (Napoles et al., 2017; Mizumoto et al., 2011; Daudaravicius, 2015). The size of our dataset (10k sentence pairs) is six times greater than that of JFLEG, which contains both

[6]We corrected spelling errors using https://github.com/barrust/pyspellchecker

Dataset	size	w/mask	w/change	L.D.
Lang-8	2.1M	-	42%	3.5
AESW	1.2M	-	39%	4.8
JFLEG	1.5k	-	86%	12.4
SMITH	10k	33%	99%	47.0

Table 3: Comparison with existing datasets. w/mask and w/change denote the percentage of source sentences with mask tokens and the percentage where the source and target sentences differ, respectively. L.D. indicates the averaged character-level Levenshtein distance between the pairs of sentences.

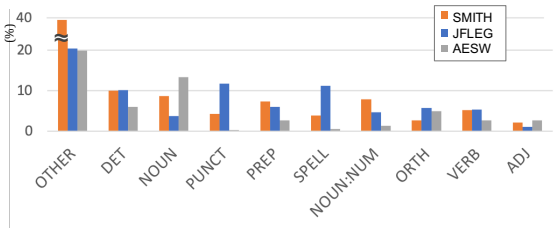

Figure 3: Comparison of the top 10 frequent errors observed in the 3 datasets.

grammatical errors and nonfluent wording. In addition, our dataset simulates significant editing—99% of the pairs have some changes between the draft and its corresponding reference, and 33% of the draft sentences contain gaps indicated by the special token <⋆>. We also measured the amount of change from the drafts X to the references Y by using the Levenshtein distance between them. A higher Levenshtein distance between the X and Y sentences in our dataset indicated more significant differences between them compared with major GEC corpora. This finding implies that our dataset emulates more drastic rephrasing.

4 Analysis of the SMITH dataset

In this section, we run extensive analyses on the sentences written by non-native workers (*draft* sentences X), and the original sentences extracted from the set of accepted papers (*reference* sentences Y). We randomly selected a set of 500 pairs from SMITH as the development set for analysis.

4.1 Error type comparison

To obtain the approximate distributions of error types between the source and target sentences, we used ERRANT (Bryant et al., 2017; Felice et al., 2016). Next, we compared them with three datasets: SMITH, AESW (the same domain as SMITH), and JFLEG (has a relatively close Levenshtein distance to SMITH). To calculate the er-

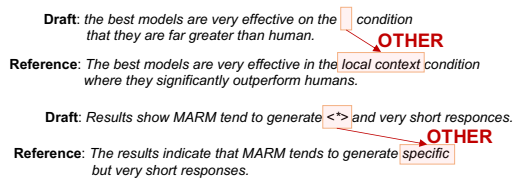

Figure 4: Examples of "OTHER" operations predicted by the ERRANT toolkit.

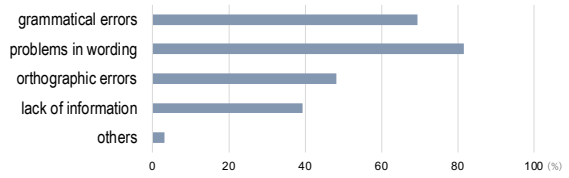

Figure 5: Result of the English experts' analyses of error types in draft sentences on our SMITH dataset. The scores show the ratio of sentences where the targeted type of errors occurred.

ror type distributions on AESW and JFLEG, we randomly sampled 500 pairs of source and target sentences from each corpus. Figure 3 shows the results of the comparison. Although all datasets contained a mix of error types and operations, the SMITH dataset included more "OTHER" operations than the other two datasets. Manual inspection of some samples of "OTHER" operations revealed that they tend to inject information missing in the draft sentence (Figure 4). This finding confirms that our dataset emphasizes a new, challenging "completion-type" task setting for writing assistance.

4.2 Human error type analysis

To understand the characteristics of our dataset in detail, an annotator proficient in English (not an author of this paper) analyzed the types of errors in the draft sentences (Figure 5). The most frequent errors were *fluency problems* (e.g., "In these *ways*" instead of "In these *methods*,")—characterized by errors in academic style and wording, which are out of the scope of traditional GEC. Another notable type of frequent error was *lack of information*, which further distinguishes this dataset from other datasets.

4.3 Human fluency analysis

We outsourced the scoring of the fluency of the given draft and reference sentence pairs to three annotators proficient in English. Nearly every draft x (94.8%) was marked as being less fluent than its corresponding reference y, confirming that obtaining high performance with our dataset requires the ability to transform rough input sentences into more fluent sentences.

Data	FRE	passive voice (%)	word repetition (%)	PPL
Draft X	45.5	34.0	33.0	1373
Reference Y	40.0	29.6	28.6	147

Table 4: Comparison of the draft and reference sentences in SMITH. FRE and PPL scores were calculated once in each sentence and then averaged over all the sentences in the development set of SMITH.

4.4 Sentence-level linguistic characteristics

We computed some sentence-level linguistic measures over the dataset sentences: Flesch Reading Ease (FRE) (Flesch, 1948), passive voice[7], word repetition, and perplexity (PPL) (Table 4).

FRE measures the *readability* of a text, namely, how easy it is to understand (higher is easier). The draft sentences consistently demonstrated higher FRE scores than their reference counterparts, which may be attributed to the latter containing more sophisticated language and technical terms.

In addition, workers tended to use the passive voice and to repeat words within a narrow span, and both those phenomenon must be avoided in academic writing. We conducted further analyses on lexical tendencies between the drafts and references (Appendix A).

Finally, we analyzed the draft and the reference sentences using PPL calculated by a 5-gram language model trained on ACL Anthology papers.[8] The higher PPL scores in the draft sentences (Table 4) suggest that they have properties unsuitable for academic writing (e.g., less fluent wording).

5 Experiments

5.1 Baseline models

We evaluated three baseline models on the SentRev task.

5.1.1 Heuristic noising and denoising model

We can access a great deal of final version academic papers. Noising and denoising approaches

[7] https://github.com/armsp/active_or_passive

[8] PPL is calculated with the implementation available in the KenLM (https://github.com/kpu/kenlm), tuned on AASC (excluding the texts used for building the SMITH).

method	original	generated
Heuristic	Besides , the recognizer successfully rejected only 15 out of 42 negative sentences .	recognizer Besides successfully , the informativeness rejected of out <⋆>
Grammatical error generation	We plan to **analyze** these direct communications **and** interaction of sentiments **expressed** in these sequences of posts .	We plan to **analysis** the direct communication interaction of sentiments **express** in these sequence of posts .
Style removal	This experiment **suggested** that there were ambiguities in these pointing gestures and **led to a redesign** of the system .	This experiment **indicated** the ambiguity found in the pointing gestures and **caused a renewal** of the system .
Entailed sentence generation	Figure 2 **illustrates the effectiveness** of different features class.	There is different feature in figure 2 .

Table 5: Examples of generated training dataset.

have gained attention in the GEC and machine translation fields (Edunov et al., 2018; Xie et al., 2018; Lichtarge et al., 2019). We combined these two factors to train baseline models on noised final version sentences.

First, we collected 4,898,146 sentences Y^{aasc} from the AASC that satisfied the following conditions: (i) not included in the SMITH dataset, (ii) not too long or too short (between 5 and 35 tokens), (iii) over 50% of the characters were alphabetic. Next, we created a training dataset $(X^{\text{aasc}}_{\text{hrst}}, Y^{\text{aasc}})$ by adding noise to Y^{aasc}.

As the simplest approach for noising, we used a set of heuristic rules by randomly deleting, replacing, and swapping words in the reference sentences. Specifically, these rules included deleting words with a probability of 0.1, replacing words with a token that appeared over 10,000 times in Y_{aasc} with a probability of 0.1, and randomly shuffling the sentence while maintaining the originally adjacent words within three words apart. Next, we randomly replaced up to 50% of the words with a <⋆> token (see Appendix B for a more detailed algorithm). This method generated 4.8M heuristically noised sentences.

Subsequently, we trained a denoising model (a mapping function from $X^{\text{aasc}}_{\text{hrst}}$ to Y^{aasc}) by using Transformer (Vaswani et al., 2017) implemented in fairseq (Ott et al., 2019). We used an Adam optimizer (Kingma and Ba, 2015) with $\alpha = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10e^{-8}$. We limited the maximum tokens per each minibatch to 3000, limited the maximum number of updates to 500,000, and used a dropout rate of 0.3. The input and output texts were tokenized and then segmented into character bigrams. We used a beam width of 5 in the decoding. This model is our first baseline model for the SentRev task (henceforth, H-ND).

5.1.2 Enc-Dec noising and denoising model

As an extension of the heuristic noising and denoising model, we changed the noising methods to better simulate the characteristics of X in SMITH than the heuristic rules in Section 5.1.1. As described in Section 4, the drafts tended to (i) contain grammatical errors, (ii) use stylistically improper wording, and (iii) lack certain words. We used the following three neural Encoder-Decoder (Enc-Dec) models to generate the synthetic draft sentences.

Grammatical error generation Here, we trained a model that introduces synthetic grammatical errors to "clean" sentences by using a "flipped" dataset from GEC (clean → erroneous). We used nonidentical (source, target) sentence pairs from the Lang-8, AESW, and JFLEG datasets.

Style removal To generate stylistically unnatural sentences in the academic domain, we used paraphrasing, which preserves a sentence's content while disregarding its style. We used the ParaNMT-50M dataset (Wieting and Gimpel, 2018), a paraphrase dataset automatically created using Enc-Dec translation. We extracted parallel sentences with annotated paraphrase scores between 0.7 and 0.95 from the ParaNMT-50M dataset and used swapped pairs of source and target sentences in the dataset.

Entailed sentence generation To simulate the missing words in the draft sentences, we trained a model that generated a sentence entailed with the given text. We extracted entailed sentence pairs

Model	BLEU	ROUGE-L	BERT-P	BERT-R	BERT-F	P	R	$F_{0.5}$	Gramm.	PPL
Draft X	9.8	46.8	75.9	78.2	77.0	-	-	-	92.9	1454
H-ND	8.2	45.0	77.0	76.1	76.5	5.4	2.9	4.6	94.1	406
ED-ND	**15.4**	**51.1**	**80.9**	**80.0**	**80.4**	21.8	**12.8**	**19.2**	96.3	**236**
GEC	11.9	49.0	80.8	79.1	79.9	**22.2**	6.2	14.6	**96.7**	414
Reference Y	-	-	-	-	-	-	-	-	96.5	147

Table 6: Results of quantitative evaluation. Gramm. denotes the grammaticality score.

Draft	The global modeling using the reinforcement learning in all documents is our work in the future .
H-ND	The global modeling **of** the reinforcement learning **using** all documents **in** our work **is** the future .
ED-ND	**In our future work , we plan to explore the use of** global modeling **for** reinforcement learning in all documents .
GEC	Global modelling using reinforcement learning in all documents is our work in the future .
Reference	The global modeling using reinforcement learning for a whole document is our future work .
Draft	Also , the above **<*>** efficiently calculated by dynamic programming .
H-ND	Also , the above **results are calculated** efficiently by dynamic programming .
ED-ND	Also , the above **probabilities are calculated** efficiently by dynamic programming .
GEC	Also , the above **is** efficiently calculated by dynamic programming .
Reference	Again , the above equation can be efficiently computed by dynamic programming .
Draft	Chart4 : relation model and gold % between KL and piason .
H-ND	**Table 1 : Charx-** relation between **gold and piason and KL** .
ED-ND	**Figure 2 : CharxDiff** relation between **model and gold standard and piason** .
GEC	Chart4 : relation model and gold % between KL and person .
Reference	Table 4 : KL and Pearson correlation between model and gold probability .

Table 7: Examples of the output from the baseline models. Bold text indicates tokens introduced by the model.

from the SNLI (Bowman et al., 2015) and the MultiNLI (Williams et al., 2018) datasets.

Random noising beam search As Xie et al. (2018) pointed out, a standard beam search often yields hypotheses that are too conservative. This tendency leads the noising models to generate synthetic draft sentences similar to their references. To address this problem, we applied the random noising beam search (Xie et al., 2018) on all three noising models. Specifically, during the beam search, we added $r\beta$ to the scores of the hypotheses, where r is a value sampled from a uniform distribution over the interval $[0, 1]$, and β is a penalty hyperparameter set to 5.

We obtained 14.6M sentence pairs of ($X^{\text{aasc}}_{\text{encdec}}$, Y^{aasc}) by applying these Enc-Dec noising models to Y^{aasc}. To train the denoising model, we used both data ($X^{\text{aasc}}_{\text{hrst}}$, Y^{aasc}) and ($X^{\text{aasc}}_{\text{encdec}}$, Y^{aasc}). The model architecture was the same as the heuristic model. This denoising model is our second baseline model (ED-ND). To facilitate research in the SentRev task, we released all the 19.6M synthetic data.[9]

Analysis of the synthetic drafts Finally, we analyzed the error type distribution of the synthetic data used for training Enc-Dec noising and denoising model with ERRANT (Figure 6). The error type distribution from the synthetic dataset had similar tendencies to the one from the development set in SMITH (real-draft). KullbackLeibler divergence between these error type distributions was 0.139. This result supports the validity of our assumption that the SentRev task is a combination of GEC, style transfer, and a completion-type task.

Table 5 shows examples of the training data generated by the noising models described in Section 5. Heuristic noising, the rule-based noising method, created ungrammatical sentences. The grammatical error generation model added grammatical errors (e.g., *plan to analyze* → *plan to analysis*). The style removal model generated stylistically unnatural sentences for the academic domain (e.g., *redesign* → *renewal*). The entailed

[9] https://github.com/taku-ito/INLG2019_SentRev

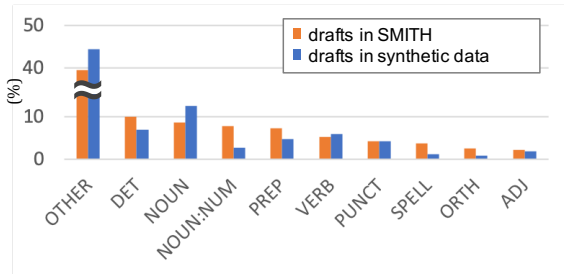

Figure 6: Comparison of the 10 most frequent error types in SMITH and synthetic drafts created by the Enc-Dec noising methods.

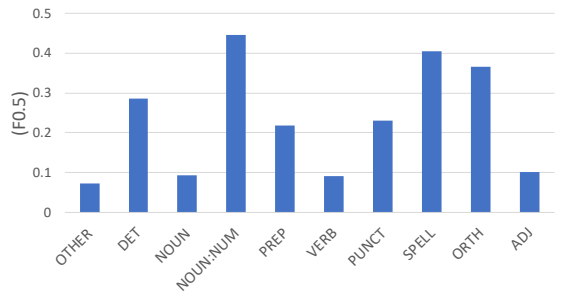

Figure 7: Performance of the ED-ND baseline model on top 10 most error types in SMITH.

sentence generation model caused a lack of information.

5.1.3 GEC model

The GEC task is closely related to SentRev. We examined the performance of the current state-of-the-art GEC model (Zhao et al., 2019) in our task. We applied spelling correction before evaluation following Zhao et al. (2019).

5.2 Evaluation metrics

The SentRev task has a very diverse space of valid revisions to a given context, which is challenging to evaluate. As one solution, we evaluated the performance from multiple aspects by using various reference and reference-less evaluation metrics. We used BLEU, ROUGE-L, and $F_{0.5}$ score, which are widely used metrics in related tasks (machine translation, style-transfer, GEC). We used nlg-eval (Sharma et al., 2017) to compute the BLEU and ROUGE-L scores and calculated $F_{0.5}$ scores with ERRANT. In addition, to handle the lexical and compositional diversity of valid revisions, we used BERT-score (Zhang et al., 2019), a contextualized embedding-based evaluation metric. Furthermore, we used two reference-less evaluation metrics: grammaticality score (Napoles et al., 2016) and PPL. Grammaticality was scored as $1 - (N_{\text{errors in sentence}}/N_{\text{tokens in sentence}})$, where the number of grammatical errors in a sentence is obtained using LanguageTools.[10] By using a language model tuned to the academic domain, we expect PPL to evaluate the stylistic validity and fluency of a complemented sentence. We favored n-gram language models over neural language models for reproducibility and calculated the score in the same manner as described in Section 4.3.

6 Results

Table 6 shows the performance of the baseline models. We observed that the ED-ND model outperforms the other models in nearly all evaluation metrics. This finding suggests that the Enc-Dec noising methods induced noise closer to real-world drafts compared with the heuristic methods.

The current state-of-the-art GEC model showed higher precision but low recall scores in $F_{0.5}$. This suggests that the SentRev task requires the model to make a more drastic change in the drafts than in the GEC task. Furthermore, the GEC model, trained in the general domain, showed the worst performance in PPL. This indicates that the general GEC model did not reflect academic writing style upon revision and that SentRev requires academic domain-aware rewriting.

Table 7 shows examples of the models' output. In the first example, the ED-ND model made a drastic change to the draft. The middle example demonstrates that our models replaced the <⋆> token with plausible words. The last example is the case where our model underperformed by making erroneous edits such as changing *"Chart4"* to *"Figure2"*, and suggesting odd content (*"relation between model and gold standard and piason"*). This may be due to having inadvertently introduced noise while generating the training datasets. Appendix C shows more examples of generated sentences. Using ERRANT, we analyzed the performance of the ED-ND baseline model by error types. The results are shown in Figure 7. Overall, typical grammatical errors such as noun number errors or orthographic errors are well corrected, but the model struggles with drastic revisions ("OTHER" type errors).

[10] https://github.com/languagetool-org/languagetool/releases/tag/v3.2

7 Related work

7.1 Writing assistance in the academic domain

Several shared tasks for assisting academic writing have been organized. The Helping Our Own (HOO) 2011 Pilot Shared Task (Dale and Kilgarriff, 2011) aimed to promote the development of tools and techniques to assist authors in writing, with a specific focus on writing within the NLP community. The Automated Evaluation of Scientific Writing (AESW) Shared Task (Daudaravicius, 2015) was organized to promote tools to help write scientific papers. The HOO dataset was created by finding errors in published papers and editing the errors, and the AESW dataset contains a collection of text extracts from published journal articles before and after proofreading. Rather than adding finishing touches to almost completed sentences, our task is to convert unfinished, rough drafts into complete sentences. In addition, these studies tackled the task of the *identification* of errors while SentRev goes further by *rewriting* the drafts.

Other corpora for revisions are available in the academic domain (Lee and Webster, 2012; Tan and Lee, 2014; Zhang et al., 2017). Thus, we provide a notable contribution by exploring the methods to create a dataset of revisions with a scalable crowdsourcing approach. By contrast, Zhang et al. (2017) recruited 60 students over 2 weeks and Lee and Webster (2012) collected data from a language learning project where over 300 tutors reviewed academic essays written by 4500 students.

7.2 Grammatical error correction

GEC is the task of correcting errors in text such as spelling, punctuation, grammar, and word choice (Ng et al., 2014; Yuan and Briscoe, 2016). GEC falls within the *editing* and *proofreading* phases of the writing process, while SentRev subsumes GEC and a broader range of text generation (e.g., increasing the fluency of the sentence and complementing missing information). Napoles et al. (2017) and Sakaguchi et al. (2016) explored fluency edits to correct grammatical errors and to make a text more "native sounding." Although this direction is similar to SentRev, our task used sentences that required many more corrections.

7.3 Style transfer

Style transfer is the task of rephrasing the text to conform to specific stylistic properties while preserving the text's original semantic content (Logeswaran et al., 2018; Prabhumoye et al., 2018). From the perspective of automatic academic writing assistance, the assistance systems are required to convert nonacademic-style drafts into academic-style drafts. This type of transfer is regarded as a subproblem in the *revising* stage of the writing process.

7.4 Text completion

The drafts in the *revising* stage may contain gaps denoted with <⋆>. This setting is similar to *text infilling* (Zhu et al., 2019), masking-based language modeling (Fedus et al., 2018; Devlin et al., 2019), or the *sentence completion task* (Zweig et al., 2012), where the models are required to replace mask tokens with plausible words. Notably, SentRev differs from such tasks because systems for these tasks are expected to keep all the original tokens unchanged and only fill the <⋆> token, with one or more other tokens.

8 Conclusion and future work

We proposed the SentRev task, where an incomplete, rough draft sentence is transformed into a more fluent, complete sentence in the academic writing domain. We created the SMITH dataset with crowdsourcing for development and evaluation of this task and established baseline performance with a synthetic training dataset. We believe that this task can increase the effectiveness of the process of academic writing. In future work, we plan to improve the information gap-filling aspect of revision by considering the surrounding context of target sentences. In addition, to develop a more holistic writing assistance tool, we plan to extend our system to be able to suggest diverse correction candidates, provide interactive assistance, and integrate translation systems.

9 Acknowledgements

We thank the Tohoku NLP laboratory members who provided us with their valuable advice. We are grateful to Benjamin Heinzerling and Marie-Josée Brassard for their feedback. We are also grateful to Masato Mita for advice on the experiments. The work of J. Suzuki was partly supported by JSPS KAKENHI Grant Number 19H04162.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of EMNLP*, pages 632–642.

Eric Brill and Robert C. Moore. 2000. An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of ACL*, pages 286–293.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of ACL*, pages 793–805.

Michael Buchman, Roberta Moore, Linda Stern, and Betsy Feist. 2000. *Power Writing: Writing with Purpose*, volume 4.

Andrew D Cohen and Amanda Brooks-Carson. 2001. Research on direct versus translated writing: Students' strategies and their results. *The Modern Language Journal*, 85(2):169–188.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of ENLG*, pages 242–249.

Vidas Daudaravicius. 2015. Automated Evaluation of Scientific Writing: AESW Shared Task Proposal. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 56–63.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of EMNLP*, pages 489–500.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better Text Generation via Filling in the _. In *Proceedings of ICLR*.

Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In *Proceedings of COLING*, pages 825–835.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221 – 233.

David Grangier and Michael Auli. 2018. QuickEdit: Editing Text & Translations by Crossing Words Out. In *Proceedings of NAACL-HLT*, pages 272–282.

Jason S. Kessler. 2017. Scattertext: a Browser-Based Tool for Visualizing how Corpora Differ. In *Proceedings of ACL System Demonstrations*, pages 85–90.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*.

John Lee and Jonathan Webster. 2012. A corpus of textual revisions in second language writing. In *Proceedings of ACL*, pages 248–252.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of NAACL-HLT*, pages 3291–3301.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content Preserving Text Generation with Attribute Controls. In *Proceedings of NeurIPS,*, pages 5103–5113.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. In *Proceedings of IJCNLP*, pages 147–155.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction. In *Proceedings of EMNLP*, pages 2109–2115.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In *Proceedings of EACL*, 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 CoNLL:Shared Task*, pages 1–14.

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 NAACL-HLT*, pages 48–53.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style Transfer Through Back-Translation. In *Proceedings of ACL*, pages 866–876.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the Goals of Grammatical Error Correction: Fluency instead of Grammaticality. *TACL*, 4:169–182.

Anthony Seow. 2002. The writing process and process writing. *Methodology in language teaching: An anthology of current practice*, pages 315–320.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation. *arXiv preprint arXiv:1706.09799*.

Bernard Susser. 1994. Process approaches in ESL/EFL writing instruction. *Journal of Second Language Writing*, 3(1):31–47.

Chenhao Tan and Lillian Lee. 2014. A corpus of sentence-level revisions in academic writing: A step towards understanding statement strength in communication. In *Proceedings of ACL*, pages 403–408.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of NIPS*, pages 5998–6008.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. In *Proceedings of ACL*, pages 451–462.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

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

Zheng Yuan and Ted Briscoe. 2016. Grammatical Error Correction using Neural Machine Translation. In *Proceedings of NAACL-HLT*, 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 ACL*, pages 1568–1578.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav. Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the NAACL-HLT*, pages 156–165.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text Infilling. *arXiv preprint arXiv:1901.00158*.

Geoffrey Zweig, John C. Platt, Christopher Meek, Christopher J. C. Burges, Ainur Yessenalina, and Qiang Liu. 2012. Computational Approaches to Sentence Completion. In *Proceedings of ACL*, pages 601–610.

A Lexical tendencies

Certain words and phrases were more frequently observed in the reference sentences than in the draft sentences, and vice-versa. Figure 8 visualizes these biases, where words more often observed in the draft sentences are plotted in the upper-left corner, and words more often observed in the references are plotted in the lower-right corner. Words observed more commonly in the drafts were: *will*, *is not*, *if*, and *I*, versus *can be*, *no*, *when*, and *they*. The contrast also includes a widely-used spelling (*data set* vs *dataset*) and common plurality (*method* vs *methods*). The plot was generated using the scattertext toolkit (Kessler, 2017).

B Heuristic noising algorithm

Algorithm 1 shows the noising algorithm in the heuristic noising method.

C Examples from the SMITH dataset and generated sentences by Baseline models

Table 8 shows examples from the SMITH dataset and the output of the baseline models. "Reference" is a sentence extracted from papers, "Draft" is written by a crowdworker and is the input for the baseline models.

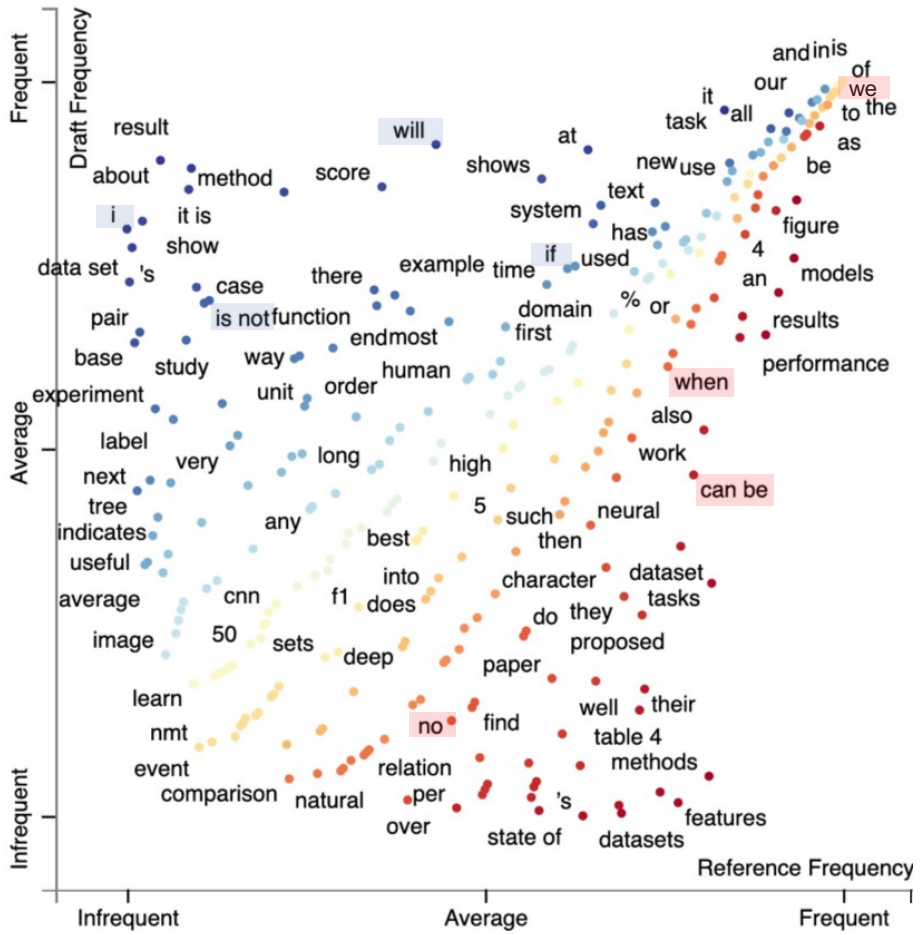

Figure 8: Characteristic words and phrases in draft sentences and reference sentences in the development set of SMITH.

Algorithm 1 Heuristic noising

INPUT: $x = \{w_0, w_1, \cdots, w_n\}$

$x = \text{delete}(x, 0.1)$
10% of the tokens in x are deleted.
$x = \text{replace}(x, 0.1)$
10% of the tokens in x are replaced with common terms in ACL.
$x = \text{permutate}(x)$
permutate the tokens in x.
$r \leftarrow \text{Uniform}(0, 0.5)$
$m = \text{int}(x.\text{length} * r)$
$c = 0$
while $c < m$ **do**
  $n \leftarrow \text{sample}(\{j \in \mathcal{N} \mid 1 \leq j \leq m - c\})$
  $(s, e) \leftarrow \text{sample}(\{n\text{-grams of } x\})$
  $x = \text{“}x_{:s-1} + \texttt{<*>} + x_{e+1:}\text{”}$
  $c = c + n$
end while
# $r \times 100\%$ of the tokens in x are masked.

Draft	By this setting , the persona is acquired from a test set popl about both turker anad model .
H-ND	By this setting , the persona is acquired from a test set both about popl anad anad model .
ED-ND	In this setting , persona is obtained from the test set popl about both Turker and model .
GEC	By this setting , the persona is acquired from a test set pool about both turkey and models .
Reference	In this setting , for both the Turker and the model , the personas come from the test set pool .
Draft	In addition to results of study until now , we add two baseline to vindicate effectiveness on our flame work .
H-ND	In addition to the results of this study , we now add two baseline methods to vindicate effectiveness on our work .
ED-ND	In addition to the results of the study until now , we add two baselines to visualize the effectiveness of our framework .
GEC	In addition to the results of study until now , we added two baseline to vindicate effectiveness on our flame work .
Reference	In addition to results of previous work , we add two baselines to demonstrate the effectiveness of our framework .
Draft	Yhe input and output <⋆> are one - hot encoding of the center word and the context word , <⋆> .
H-ND	The input and output are one - hot encoding of the center word and the context word , respectively .
ED-ND	The input and output layers are one - hot encoding of the center word and the context word , respectively .
GEC	Yhe input and output are one - hot encoding of the center word and the context word , .
Reference	The input and output layers are centre word and context word one - hot encodings , respectively .
Draft	I registered the vocabulary sizes of encorder and decorder as 150 K and 50 K each other .
H-ND	I registered the vocabulary sizes of decorder and encorder as 150 K and each other .
ED-ND	We registered the vocabulary sizes of the encoder and decoder as 150 K and 50 K respectively .
GEC	I registered the vocabulary sizes of encoder and recorder as 150 K and 50 K for each other .
Reference	In this experiment , we set the vocabulary size on the encoder and decoder sides to 150 K and 50 K , respectively .
Draft	They add the new class image generated by generator and classfy them .
H-ND	They add the new image class generated by the generator and classfy them .
ED-ND	They add a new class of images generated by the generator and classify them .
GEC	They add a new class image generated by generator and classify them .
Reference	They add a new class of images that are generated by the generator and classify them .
Draft	The chart 3 shows performance of multi input correction against sub groups with different number of witnesses .
H-ND	Table 3 shows the performance of multi - chart correction against different input groups with different number of witnesses .
ED-ND	Figure 3 shows the performance of multiple input correction against subgraphs with different number of witnesses .
GEC	chart 3 shows performance of multi input correction against sub groups with different number of witnesses .
Reference	Figure 3 presents the performance of multi - input correction on subgroups with different number of witnesses .
Draft	It is vindicated that InferSent accomplishes the most <⋆> result regarding SentEval task .
H-ND	It is vindicated that InferSent accomplishes the most relevant result regarding the SentEval task .
ED-ND	It is vindicated that InferSent accomplishes the most important result regarding the SentEval task .
GEC	It is vindicated that InferSent accomplishes the most results regarding SentEval task .
Reference	InferSent has been shown to achieve state - of - the - art results on the SentEval tasks .
Draft	Our proposal model can get both long - term dependence and local information well .
H-ND	Our proposal can get both long - term and local information as well .
ED-ND	Our proposed model can capture both long - term dependencies and local information well .
GEC	Our proposal model can get both long - term dependence and local information well .
Reference	Our proposed model can both capture long - term dependencies and local information well .

Table 8: Further examples of draft, reference, and the baseline models' output.