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

Takumi Ito\textsuperscript{1,2}, Tatsuki Kuribayashi\textsuperscript{1,2}, Hayato Kobayashi\textsuperscript{3,4}, Ana Brassard\textsuperscript{4,1}, Masato Hagiwara\textsuperscript{5}, Jun Suzuki\textsuperscript{1,4} and Kentaro Inui\textsuperscript{1,4}

1: Tohoku University, 2: Langsmith Inc., 3: Yahoo Japan Corporation, 4: RIKEN, 5: Octanove Labs LLC
The writing process

FIRST DRAFT: “Model have good results.”

Revising “Our model show good result in this task.”  “Our model shows a excellent performance in this task.”

Editing “Our model shows good results in this task.”  “Our model shows a excellent performance in this task.”

Proofreading “Our model shows excellent performance in this task.”

FINAL VERSION: “Our model shows excellent performance in this task.”
Automatic writing assistance

- insufficient fluidity
- awkward style
- collocation errors
- missing words

- grammatical errors
- spelling errors

**FIRST DRAFT:** “Model have good results.”

**Revising**

“Our model show good result in this task.”

“Our model shows a excellent perfomance in this task.”

**Editing**

“Our model shows good results in this task.”

“Our model shows a excellent perfomance in this task.”

**Proofreading**

“Our model shows excellent performance in this task.”

**FINAL VERSION:**

“Our model shows excellent performance in this task.”
Automatic writing assistance

FIRST DRAFT: “Model have good results.”

Revising “Our model show good result in this task.” “Our model shows a excellent perfomance in this task.”

EXISTING STUDIES

Editing “Our model shows good results in this task.” “Our model shows a excellent performance in this task.”

Proofreading “Our model shows excellent performance in this task.”

FINAL VERSION: “Our model shows excellent performance in this task.”

✗ insufficient fluidity
✗ awkward style
✗ collocation errors
✗ missing words

✓ grammatical errors
✓ spelling errors

Grammatical error correction (GEC)

2019/10/29
Automatic writing assistance

- insufficient fluidity
- awkward style
- collocation errors
- missing words

✓ grammatical errors
✓ spelling errors

Grammatical error correction (GEC)

**Sentence-level revision (SentRev)**

**FIRST DRAFT:** “Model have good results.”

**OUR FOCUS**

**Revising**

“Our model show good result in this task.”

“Our model shows a excellent performance in this task.”

**Editing**

“Our model shows good results in this task.”

“Our model shows a excellent performance in this task.”

**Proofreading**

“Our model shows excellent performance in this task.”

**FINAL VERSION:**

“Our model shows excellent performance in this task.”
Proposed Task: Sentence-level Revision

- input: early-stage draft sentence
  - has errors (e.g., collocation errors)
  - has Information gaps (denoted by <*>)

- output: final version sentence
  - error-free
  - correctly filled-in sentence

Our approach idea is <*> at read pattern of normal human.

The idea of our approach derives from the normal human reading pattern.
Proposed Task: Sentence-level Revision

Our approach idea is <*> at read pattern of normal human.

The idea of our approach derives from the normal human reading pattern.

- input: early-stage draft sentence
  - has errors (e.g., collocation errors)
  - has Information gaps (denoted by <*>)

- output: final version sentence
  - error-free
  - correctly filled-in sentence

- issue: lack of evaluation resource
Our contributions

Created an evaluation dataset for SentRev
- Set of Modified Incomplete Technical paper sentences (SMITH)

Analyzed the characteristics of the dataset

Established baseline scores for SentRev
Evaluation Dataset Creation

**Goal**: collect pairs of draft sentence and final version

![Diagram showing draft and final versions with model results]

- **Draft**: *Our model <-> results*
- **Final**: *Our model shows competitive results*
Evaluation Dataset Creation

**Goal**: collect pairs of draft sentence and final version

- Our model $\leftrightarrow$ results
- Our model shows competitive results

**Straight-forward approach**:
Experts modify collected drafts to final version

**limitation**: early-stage draft sentences are not usually publicly available

**Note**: We can access plenty of final version sentences
Evaluation Dataset Creation

**Goal:** collect pairs of draft sentence and final version

Our model $\leftrightarrow$ results

**Straight-forward approach:**
Experts modify collected drafts to final version

Our approach:
create draft sentences from final version sentences

Our model shows competitive results
Crowdsourcing Protocol for Creating an Evaluation Dataset

Our approach:
create draft sentences from final version sentences

1. automatically translate the final sentence into Japanese
2. Japanese native workers translate into English

Our model shows competitive results
Crowdsourcing Protocol for Creating an Evaluation Dataset

Our approach:
create draft sentences from final version sentences

1. automatically translate the final sentence into Japanese

2. Japanese native workers translate into English

Our model shows competitive results

Our model <**> results

insert <**> where workers could not think of a good expression

ACL Anthology

final version

私たちのモデルは匹敵する結果を示しました。
### Statistics

<table>
<thead>
<tr>
<th>Dataset</th>
<th>size</th>
<th>w/&lt;<em>&gt;</em></th>
<th>w/change</th>
<th>Levenshtein distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lang-8</td>
<td>2.1M</td>
<td>-</td>
<td>42%</td>
<td>3.5</td>
</tr>
<tr>
<td>AESW</td>
<td>1.2M</td>
<td>-</td>
<td>39%</td>
<td>4.8</td>
</tr>
<tr>
<td>JFLEG</td>
<td>1.5K</td>
<td>-</td>
<td>86%</td>
<td>12.4</td>
</tr>
<tr>
<td>SMITH</td>
<td>10K</td>
<td>33%</td>
<td>99%</td>
<td>47.0</td>
</tr>
</tbody>
</table>

w/<*>*: percentage of source sentences with <*>  
w/change: percentage where the source and target sentences differ

- collected 10,804 pairs  
- SMITH simulates significant editing  
- Larger Levenshtein distance ⇒ more drastic editing
Examples of SMITH

**draft:** I research the rate of workable SQL <*> at the generated result.

**final:** We study the percentage of executable SQL queries in the generated results.

**draft:** For <*/, we used Adam using weight decay and gradient clipping.

**final:** We used Adam with a weight decay and gradient clipping for optimization.

**draft:** In the model architecture, as shown in Figure 1, it is based on an AE and GAN.

**final:** The model architecture, as illustrated in figure 1, is based on the AE and GAN.
Examples of SMITH

(1) Wording problems

draft:  I research the rate of workable SQL <*> at the generated result.

final:  We study the percentage of executable SQL queries in the generated results.

draft:  For <*>, we used Adam using weight decay and gradient clipping.

final:  We used Adam with a weight decay and gradient clipping for optimization.

draft:  In the model architecture, as shown in Figure 1, it is based an AE and GAN.

final:  The model architecture, as illustrated in figure 1, is based on the AE and GAN.
Examples of SMITH

(1) Wording problems

draft:  I research the rate of workable SQL <*> at the generated result.

final:  We study the percentage of executable SQL queries in the generated results.

draft:  For <*> , we used Adam using weight decay and gradient clipping.

final:  We used Adam with a weight decay and gradient clipping for optimization.

draft:  In the model architecture, as shown in Figure 1 , it is based on an AE and GAN.

final:  The model architecture, as illustrated in figure 1 , is based on the AE and GAN.
Examples of SMITH

(2) Information gaps

draft: I research the rate of workable SQL <*> at the generated result.

final: We study the percentage of executable SQL queries in the generated results.

draft: For <*/, we used Adam using weight decay and gradient clipping.

final: We used Adam with a weight decay and gradient clipping for optimization.

draft: In the model architecture, as shown in Figure 1, it is based an AE and GAN.

final: The model architecture, as illustrated in figure 1, is based on the AE and GAN.
Examples of SMITH

(2) Information gaps

draft:  I research the rate of workable SQL \(<*>\) at the generated result.

final:  We study the percentage of executable SQL queries in the generated results.

draft:  For \(<*>\), we used Adam using weight decay and gradient clipping.

final:  We used Adam with a weight decay and gradient clipping for optimization.

draft:  In the model architecture, as shown in Figure 1, it is based an AE and GAN.

final:  The model architecture, as illustrated in figure 1, is based on the AE and GAN.
Examples of SMITH

(3) Spelling and grammatical errors

draft: I research the rate of workable SQL \(<*>\) at the generated result.

final: We study the percentage of executable SQL queries in the generated results.

draft: For \(<*>\), we used Adam using weight decay and gradient clipping.

final: We used Adam with a weight decay and gradient clipping for optimization.

draft: In the model architecture, as shown in Figure 1, it is based on AE and GAN.

final: The model architecture, as illustrated in figure 1, is based on the AE and GAN.
A great deal of research has been carried out in grammar error correction. Many studies have been conducted in this area. Baseline models were built to revise draft versions into final versions.

- **Built baseline revision models (draft ⇒ final version)**
  - Training data: generated synthetic data with noising methods

- **Evaluated the performance on SMITH**
  - Using various reference and reference-less evaluation metrics
Noising and Denoising

Noising: automatically generate drafts from the final versions

A great deal of research has been carried out in grammar error correction.

many study $<$*> in grammar error correction

ACL Anthology

sample

Nosing methods

final version
A great deal of research has been carried out in grammar error correction.

Denoising models (Baseline models)

Denoising: generate final versions from the drafts

many study <> in grammar error correction

many study <> in grammar error correction

draft

final version

Nosing methods

ACL Anthology

sample
it is not surprisingly that the random policy have the worst performing.

we see the same on larger data.

Figure 2 illustrates effectiveness

perplexity indicates a <*> model.

Grammatical error generation

Style removal

Entailed sentence generation

Heuristic

it is not surprising that the random policy has the worst performance.

we observe a similar trend on larger datasets.

Figure 2 illustrates the effectiveness of different features.

lower perplexity indicates a better model.
Noising methods

Drafts:

It is not surprisingly that the random policy have the worst performing.

Noising methods:

Grammatical error generation

Final versions:

It is not surprising that the random policy has the worst performance.

We see the same on larger data.

We observe a similar trend on different features.

Train Enc-Dec noising model (clean ⇒ erroneous) using Lang8 [Mizumoto+ 11], AESW [Daudaravicius+ 15], and JFLEG [Napoles+ 17].

Perplexity indicates a better model.

Lower perplexity indicates a better model.
Noising methods

Drafts

- it is not surprisingly that the random policy has the worst performing.
- we see the same on larger data.

Noising methods

- Grammatical error generation
- Style removal

- we observe a similar trend on larger datasets.

Final versions

- it is not surprising that the random policy has the worst performance.

Figure 2 illustrates the effectiveness of different features.

Train Enc-Dec noising model (academic ⇝ non-academic) using the ParaNMT-50M dataset [Wieting+18]
it is not surprisingly that the random policy has the worst performance.

we see the same on larger data.

Figure 2 illustrates the effectiveness of different features.

Figure 2 illustrates the effectiveness of different features.

Heuristic

lower perplexity indicates a better model.

perplexity indicates a <*> model.

Entailed sentence generation

Grammatical error generation

train Enc-Dec noising model (⇒ entailed sentence) using SNLI [Bowman+ 15], MultiNLI [Williams+ 18]
Noising methods

Drafts

- it is not surprisingly that the random policy has the worst performing.

- we see the same on larger data.

Noising methods

- Grammatical error generation

- Style removal

Final versions

- it is not surprising that the random policy has the worst performance.

- we observe a similar trend on larger datasets.

Heuristic noising rules:
randomly deleting, replacing with <*> or common terms, and swapping

Perplexity indicates a <*> model.

Lower perplexity indicates a better model.
Baseline models

- Noising and Denoising models
  - Heuristic noising and denoising model (H-ND)
    - Rule-based Heuristic noising (e.g., random token replacing)
  - Enc-Dec noising and denoising model (ED-ND)
    - Rule-based Heuristic noising
      + trained error generation models (e.g., grammatical error generation)

- SOTA GEC model [Zhao+ 19]
Experiment settings

- Noising and Denoising Model architecture
  - Transformer [Vaswani+ 17]
  - Optimizer: Adam with $\alpha = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-8}$

- Evaluation metrics
  - BLEU
  - ROUGE-L
  - F0.5
  - BERTscore [Zhang+ 19]
  - Grammaticality score [Napoles+ 16]: $1 - \frac{\text{#errors in sent}}{\text{#tokens in sent}}$
  - Perplexity (PPL): 5-gram LM trained on ACL Anthology papers
Results

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>ROUGE-L</th>
<th>BERT-P</th>
<th>BERT-R</th>
<th>BERT-F</th>
<th>P</th>
<th>R</th>
<th>F0.5</th>
<th>Gramm.</th>
<th>PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draft X</td>
<td>9.8</td>
<td>46.8</td>
<td>75.9</td>
<td>78.2</td>
<td>77.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>92.9</td>
<td>1454</td>
</tr>
<tr>
<td>H-ND</td>
<td>8.2</td>
<td>45.0</td>
<td>77.0</td>
<td>76.1</td>
<td>76.5</td>
<td>5.4</td>
<td>2.9</td>
<td>4.6</td>
<td>94.1</td>
<td>406</td>
</tr>
<tr>
<td>ED-ND</td>
<td>15.4</td>
<td>51.1</td>
<td>80.9</td>
<td>80.0</td>
<td>80.4</td>
<td>21.8</td>
<td>12.8</td>
<td>19.2</td>
<td>96.3</td>
<td>236</td>
</tr>
<tr>
<td>GEC</td>
<td>11.9</td>
<td>49.0</td>
<td>80.8</td>
<td>79.1</td>
<td>79.9</td>
<td>22.2</td>
<td>6.2</td>
<td>14.6</td>
<td>96.7</td>
<td>414</td>
</tr>
<tr>
<td>Reference Y</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>96.5</td>
<td>147</td>
</tr>
</tbody>
</table>

- ED-ND model outperforms the other models
  - the HD-ND noising methods induced noise closer to real-world drafts
- SOTA GEC model showed higher precision but low recall
  - the GEC model is conservative
### Examples of the baseline models’ output

<table>
<thead>
<tr>
<th>Draft</th>
<th>Yhe input and output &lt;<em>&gt; are one-hot encoding of the center word and the context word, &lt;</em>&gt;.</th>
</tr>
</thead>
<tbody>
<tr>
<td>H-ND</td>
<td>The input and output are one-hot encoding of the center word and the context word, respectively.</td>
</tr>
<tr>
<td>ED-ND</td>
<td>The input and output layers are one-hot encoding of the center word and the context word, respectively.</td>
</tr>
<tr>
<td>GEC</td>
<td>Yhe input and output are one-hot encoding of the center word and the context word.</td>
</tr>
<tr>
<td>Reference</td>
<td>The input and output layers are center word and context word one-hot encodings, respectively.</td>
</tr>
</tbody>
</table>

ED-ND models replaced the <*> token with plausible words.
Analysis: error types of drafts in SMITH & training data

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.

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 F0.5 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 F0.5 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 $(N \text{ errors in sentence} / N \text{ tokens in sentence})$, where the number of grammatical errors in a sentence is obtained using LanguageTools.

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 F0.5. 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).

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

2019/10/29
Conclusions

- proposed the SentRev task
  - Input: a incomplete, rough draft sentence
  - Output: a more fluent, complete sentence in the academic domain.

- created the SMITH dataset with crowdsourcing for development and evaluation of this task

- established baseline performance with a synthetic training dataset
  - training dataset available at the same link as above
## Criteria for evaluating crowdworkers

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Judgment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working time is too short (&lt; 2 minutes)</td>
<td>Reject</td>
</tr>
<tr>
<td>All answers are too short (&lt; 4 words)</td>
<td>Reject</td>
</tr>
<tr>
<td>No answer ends with “.” or “?”</td>
<td>Reject</td>
</tr>
<tr>
<td>Contain identical answers</td>
<td>Reject</td>
</tr>
<tr>
<td>Some answers have Japanese words</td>
<td>Reject</td>
</tr>
<tr>
<td>No answer is recognized as English</td>
<td>Reject</td>
</tr>
<tr>
<td>Some answers are too short (&lt; 4 words)</td>
<td>-2 points</td>
</tr>
<tr>
<td>Some answers use fewer than 4 kinds of words</td>
<td>-2 points</td>
</tr>
<tr>
<td>Too close to automatic translation (20 &lt;= L.D. &lt;= 30)</td>
<td>-0.5 points/ans</td>
</tr>
<tr>
<td>Too close to automatic translation (10 &lt;= L.D. &lt;= 20)</td>
<td>-1.5 points/ans</td>
</tr>
<tr>
<td>Too close to automatic translation (L.D. &lt;= 10)</td>
<td>Reject</td>
</tr>
<tr>
<td>All answers end with “.” or “?”</td>
<td>+1 points</td>
</tr>
<tr>
<td>Some answers have &lt;スター&gt;</td>
<td>+1 points</td>
</tr>
<tr>
<td>All answers are recognized as English</td>
<td>+1 points</td>
</tr>
</tbody>
</table>

- filtered the crowdworkers' answers using the criteria
- accepted answers with score 0 or higher
3.2 Statistics

dataset and confirms the quality of S

In this section, we run extensive analyses on the

Comparison of the top 10 frequent errors observed in the 3 datasets

SMITH included more “OTHER” than the other two datasets

<table>
<thead>
<tr>
<th>Error Type</th>
<th>SMITH</th>
<th>JFLEG</th>
<th>AESW</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTHER</td>
<td>40%</td>
<td>20%</td>
<td>10%</td>
</tr>
<tr>
<td>DET</td>
<td>15%</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>NOUN</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>PUNCT</td>
<td>5%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>PREP</td>
<td>5%</td>
<td>5%</td>
<td>5%</td>
</tr>
<tr>
<td>SPELL</td>
<td>3%</td>
<td>3%</td>
<td>3%</td>
</tr>
<tr>
<td>NOUN:NUM</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
<tr>
<td>ORTH</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
<tr>
<td>VERB</td>
<td>2%</td>
<td>2%</td>
<td>2%</td>
</tr>
<tr>
<td>ADJ</td>
<td>1%</td>
<td>1%</td>
<td>1%</td>
</tr>
</tbody>
</table>
Examples of “OTHER” in SMITH

**Draft:** The best models are very effective on the condition that they are far greater than human.

**Reference:** The best models are very effective in the local context condition where they significantly outperform humans.

**Draft:** Results show MARM tend to generate <*> and very short responses.

**Reference:** The results indicate that MARM tends to generate specific but very short responses.

SMITH emphasizes “completion-type” task setting for writing assistance.