An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction

Shun Kiyono$^{1,2}$ Jun Suzuki$^{2,1}$ Masato Mita$^{1,2}$ Tomoya Mizumoto$^{1,2,*}$ Kentaro Inui$^{2,1}$

$^1$RIKEN Center for Advanced Intelligence Project $^2$Tohoku University

{shun.kiyono, masato.mita, tomoya.mizumoto}@riken.jp;
{jun.suzuki, inui}@ecei.tohoku.ac.jp

Code available at https://github.com/butsugiri/gec-pseudodata
Grammatical Error Correction (GEC)

• Input: sentence with grammatical error
• Output: sentence without grammatical error
• GEC is commonly tackled as Machine Translation

I follows his advice

I follow his advice

Model (e.g. Encoder-Decoder)
GEC data is “low-resource”

- Amount of parallel data is limited in GEC (~2M)
- More data is important for better performance

- Pseudo data generation is currently the best method to increase GEC data
  - Adopted by most teams in BEA 2019 Shared Task

Table 2: German-English IWSLT results for training corpus size of 100k words and 3.2M words (full corpus). Mean and standard deviation of three training runs reported.

4.3 NMT Systems

We train neural systems with Nematus (Sennrich et al., 2017b). Our baseline mostly follows the settings in (Koehn and Knowles, 2017); we use adam (Kingma and Ba, 2015) and perform early stopping based on dev set BLEU. We express our batch size in number of tokens, and set it to 4000 in the baseline (comparable to a batch size of 80 sentences used in previous work).

We subsequently add the methods described in section 3, namely the bideep RNN, label smoothing, dropout, tied embeddings, layer normalization, changes to the BPE vocabulary size, batch size, model depth, regularization parameters and learning rate. Detailed hyperparameters are reported in Appendix A.

5 Results

Table 2 shows the effect of adding different methods to the baseline NMT system, on the ultra-low data condition (100k words of training data) and the full IWSLT 14 training corpus (3.2M words). Our “mainstream improvements” add around 6–7 BLEU in both data conditions.

In the ultra-low data condition, reducing the BPE vocabulary size is very effective (+4.9 BLEU). Reducing the batch size to 1000 token results in a BLEU gain of 0.3, and the lexical model yields an additional +0.6 BLEU. However, aggressive (word) dropout (6 BLEU) and tuning other hyperparameters (+0.7 BLEU) has a stronger effect than the lexical model, and adding the lexical model (9) on top of the optimized configuration (8) does not improve performance. Together, the adaptations to the ultra-low data setting yield 9.4 BLEU (7.2 + 16.6). The model trained on full IWSLT data is less sensitive to our changes (31.9 ± 32.8 BLEU), and optimal hyperparameters differ depending on the data condition. Subsequently, we still apply the hyperparameters that were optimized to the ultra-low data condition (8) 5 beam search results reported by Wiseman and Rush (2016).

6 p = 0.3 for dropping words; p = 0.5 for other dropout.

Figure from [Sennrich and Zhang 2019]
Training with Genuine Data only

“Genuine” Data (e.g. Lang-8)

Source Sentence

Target Sentence

Model

Training
Training with Pseudo Data

"Genuine" Data

Pseudo Data

Model

Training

Set of grammatical sentences

We have extensive amount of such data (e.g. from the Web)

Problem Solved?

Unfortunately No...

November 18, 2019
RIKEN AIP / Tohoku University
Problem: Lack of Consensus

- We have **three questions** regarding the pseudo data:
  
  Q1: Choice for seed corpus?
  Q2: Methods for generating pseudo data?
  Q3: How to use pseudo data during training?

- GEC research community **lacks consensus 😞**
- Our aim: Find **settings that consistently improve performance** of GEC model
Q1: Choice for Seed Corpus?

“Genuine” Data

Seed Corpus (e.g. Wikipedia)

Pseudo Data

Pseudo Data Generation Method

Model

Training
Q1: Choice for Seed Corpus?

• Numerous options exist:
  • Wikipedia, 1-billion word benchmark, BookCorpus, etc
  • [Zhao+2019] uses 1-billion word LM benchmark
  • [Xie+2018] uses NYT corpus
  • [Grundkiewicz+2019] uses News Crawl

• What kind of corpus is suitable for GEC?

• We compare following three corpora:
  • Simple Wikipedia
  • Wikipedia
  • LDC Gigaword

  Grammatical complexity is different
  Texts are cleaner in Gigaword
Q2: Methods for Generating Pseudo Data?

"Genuine" Data

Model

Training

Pseudo Data Generation Method

Seed Corpus (e.g. Wikipedia)

Pseudo Data
Q2: Methods for Generating Pseudo Data?

Original: At the institute, she introduced tissue culture methods that she had learned in the U.S.

Backtrans (noisy): At institute, she introduced tissue culture methods that she learned in U.S.

DirectNoise: \( \text{mask} \) the the \( \text{mask} \) \( \text{mask} \) \( \text{mask} \) \( \text{mask} \) tissue \( \text{mask} \) culture \( \text{mask} \) methods, she \( \text{mask} \) the \( \text{mask} \) the \( \text{mask} \) \( \text{mask} \) \( \text{mask} \)

We compare two methods

- **Backtrans (noisy)** [Xie+2018]
  - Data is generated by back-translation

- **DirectNoise** [Zhao+2019]
  - Data is generated by adding synthetic noise

- Please read our paper for details
Q3: How to use pseudo data during training?

"Genuine" Data

Pseudo Data

Model

Training

Seed Corpus (e.g. Wikipedia)

Pseudo Data Generation Method

Pseudo Data
Q3: How to use pseudo data during training?

**Joint Training (JOINT)**

Pre-training + Fine-tuning (PRETRAIN)

Which one performs better?
Experimental Configuration and Datasets

- We adapt “standard” configurations
  - Model: Transformer (Big) [Vaswani+2017]
  - Optimizer: Adam (for pretrain) and Adafactor (for fine-tuning)

- Dataset
  - BEA-2019 dataset (train/valid/test) [Bryant+2019]
  - CoNLL2014 (test) [Ng+2014]
Experiment 1: Choice for Seed Corpus

• Settings: **JOINT**

<table>
<thead>
<tr>
<th>Method</th>
<th>Seed Corpus $T$</th>
<th>Prec.</th>
<th>Rec.</th>
<th>$F_{0.5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>N/A</td>
<td>46.6</td>
<td>23.1</td>
<td>38.8</td>
</tr>
</tbody>
</table>

**A1: Use Gigaword**

• Seed corpus has **minor influence on $F_{0.5}$ score**
• **Gigaword** is an ideal option
  • **clean text** is more important than **domain**?
Experiment 2: Utilization of Pseudo Data

- Settings: Wikipedia as seed corpus

- If amount of pseudo data \(\approx\) genuine data
  \(\Rightarrow\) **PRETRAIN** and **JOINT** are competitive
Experiment 2: Utilization of Pseudo Data

• Settings: Wikipedia as seed corpus

- Increasing amount of pseudo data improves the performance in PRETRAIN
- performance does not improve in JOINT
  • Pseudo data becomes dominant in JOINT
Experiment 3: More Pseudo Data

- **BACKTRANS (NOISY)** significantly outperforms **DIRECTNOISE**

**A2&A3: PRETRAIN+BACKTRANS (NOISY) is effective**
Summary

A1: Use Gigaword

A2&A3: PRETRAIN+BACKTRANS (NOISY) is effective
## Comparison to Current Top Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Ensemble</th>
<th>CoNLL-2014 (M(^2) scorer)</th>
<th>BEA-test (ERRANT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>Chollampatt and Ng (2018)</td>
<td>✓</td>
<td>65.5</td>
<td>33.1</td>
</tr>
<tr>
<td>Junczys-Dowmunt et al. (2018)</td>
<td>✓</td>
<td>61.0</td>
<td>40.2</td>
</tr>
<tr>
<td>Lichtarge et al. (2019)</td>
<td>✓</td>
<td>74.8</td>
<td>58.7</td>
</tr>
<tr>
<td>Zhao et al. (2019)</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Previously Published Results**

**LARGE\text{PRETRAIN} Results**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>CoNLL-2014 (M(^2) scorer)</th>
<th>BEA-test (ERRANT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>LARGE\text{PRETRAIN}</td>
<td>67.9</td>
<td>44.1</td>
<td>61.3</td>
</tr>
<tr>
<td>LARGE\text{PRETRAIN}+SSE+R2L</td>
<td>-</td>
<td>72.1</td>
<td>61.8</td>
</tr>
<tr>
<td>LARGE\text{PRETRAIN}+SSE+R2L+SED</td>
<td>✓</td>
<td>73.3</td>
<td>44.2</td>
</tr>
</tbody>
</table>
(1) Strong Single Model Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Ensemble</th>
<th>CoNLL-2014 (M^2 scorer)</th>
<th>BEA-test (ERRANT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
</tr>
<tr>
<td>Chollampatt and Ng (2018)</td>
<td>✓</td>
<td>65.5</td>
<td>33.1</td>
</tr>
<tr>
<td>Junczys-Dowmunt et al. (2018)</td>
<td>✓</td>
<td>61.9</td>
<td>40.2</td>
</tr>
<tr>
<td>Lichtarge et al. (2019)</td>
<td>✓</td>
<td>66.7</td>
<td>43.9</td>
</tr>
<tr>
<td>Zhao et al. (2019)</td>
<td>✓</td>
<td>71.6</td>
<td>38.7</td>
</tr>
<tr>
<td>Grundkiewicz et al. (2019)</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>LARGEPretrain</strong></td>
<td></td>
<td>67.9</td>
<td>44.1</td>
</tr>
<tr>
<td><strong>LARGEPretrain+SSE+R2L</strong></td>
<td>✓</td>
<td>72.4</td>
<td>46.1</td>
</tr>
<tr>
<td><strong>LARGEPretrain+SSE+R2L+SED</strong></td>
<td>✓</td>
<td><strong>73.3</strong></td>
<td>44.2</td>
</tr>
</tbody>
</table>

Our best **single model** outperforms the **other ensemble models** except for [Grundkiewicz+2019]
## (2) Additional Techniques Improve Result

<table>
<thead>
<tr>
<th>Model</th>
<th>Ensemble</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F(_{0.5})</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F(_{0.5})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chollampatt and Ng (2018)</td>
<td>✓</td>
<td>65.5</td>
<td>33.1</td>
<td>54.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Junczys-Dowmunt et al. (2018)</td>
<td>✓</td>
<td>61.9</td>
<td>40.2</td>
<td>55.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Lichtarge et al. (2019)</td>
<td>✓</td>
<td>66.7</td>
<td>43.9</td>
<td>60.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhao et al. (2019)</td>
<td>✓</td>
<td>71.6</td>
<td>38.7</td>
<td>61.2</td>
<td>72.3</td>
<td>60.1</td>
<td>69.5</td>
</tr>
<tr>
<td>✓</td>
<td>64.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SSE: Synthetic Spelling Error**

**SED: Sentence-level Error Detection**

**R2L: Right-to-Left Reranking**

**Ensemble of 4 Models**

Our best model achieves the best performance on **CoNLL2014** (\(F\(_{0.5}\)=65.0) and **BEA-test** (\(F\(_{0.5}\)=70.2)
Conclusions

• Investigated 3 questions regarding incorporating pseudo data into GEC model.

Q1: Choice for seed corpus?
Q2: Methods for generating pseudo data?
Q3: How to use pseudo data during training?

• Discovered settings suitable (LARGEPRETRAIN)
  • justified by SOTA performance on benchmark datasets
• Code and pretrained model are available