# Improving Neural Text Normalization with Data Augmentation at Character- and Morphological Levels

**Itsumi Saito**[1] **Jun Suzuki**[2] **Kyosuke Nishida**[1] **Kugatsu Sadamitsu**[1*]
**Satoshi Kobashikawa**[1] **Ryo Masumura**[1] **Yuji Matsumoto**[3] **Junji Tomita**[1]
[1]NTT Media Intelligence Laboratories, [2]NTT Communication Science Laboratories
[3]Nara Institute of Science and Technology
{saito.itsumi, suzuki.jun, nishida.kyosuke}@lab.ntt.co.jp,
{masumura.ryo, kobashikawa.satoshi, tomita.junji}@lab.ntt.co.jp
matsu@is.naist.jp, k.sadamitsu.ic@future.co.jp

## Abstract

In this study, we investigated the effectiveness of augmented data for encoder-decoder-based neural normalization models. Attention based encoder-decoder models are greatly effective in generating many natural languages. In general, we have to prepare for a large amount of training data to train an encoder-decoder model. Unlike machine translation, there are few training data for text-normalization tasks. In this paper, we propose two methods for generating augmented data. The experimental results with Japanese dialect normalization indicate that our methods are effective for an encoder-decoder model and achieve higher BLEU score than that of baselines. We also investigated the oracle performance and revealed that there is sufficient room for improving an encoder-decoder model.

## 1 Introduction

Text normalization is an important fundamental technology in actual natural language processing (NLP) systems to appropriately handle texts such as those for social media. This is because social media texts contain non-standard texts, such as typos, dialects, chat abbreviations[1], and emoticons; thus, current NLP systems often fail to correctly analyze such texts (Huang, 2015; Sajjad et al., 2013; Han et al., 2013). Normalization can help correctly analyze and understand these texts.

One of the most promising conventional approaches for tackling text normalizing tasks is using statistical machine translation (SMT) techniques (Junczys-Dowmunt and Grundkiewicz, 2016; Yuan and Briscoe, 2016), in particular, utilizing the Moses toolkit (Koehn et al., 2007). In recent years, encoder-decoder models with an attention mechanism (Bahdanau et al., 2014) have made great progress regarding many NLP tasks, including machine translation (Luong et al., 2015; Sennrich et al., 2016), text summarization (Rush et al., 2015) and text normalization (Xie et al., 2016; Yuan and Briscoe, 2016; Ikeda et al., 2017). We can also simply apply an encoder-decoder model to text normalization tasks. However, it is well-known that encoder-decoder models often fail to perform better than conventional methods when the availability of training data is insufficient. Unfortunately, the amount of training data for text normalization tasks is generally relatively small to sufficiently train encoder-decoder models. Therefore, data utilization and augmentation are important to take full advantage of encoder-decoder models. Xie et al. (2016) and Ikeda et al. (2017) reported on improvements of data augmentation in error correction and variant normalization tasks, respectively.

Following these studies, we investigated data-augmentation methods for neural normalization. The main difference between the previous studies and this study is the method of generating augmentation data. Xie et al. (2016) and Ikeda et al. (2017) used simple morphological-level or character-level hand-crafted rules to generate augmented data. These predefined rules work well if we have sufficient prior knowledge about the target text-normalization task. However, it is difficult to cover all error patterns by simple rules and predefine the error patterns with certain text normalization tasks, such as dialect normalization whose error pattern varies from region to

*Present affiliation: Future Architect,Inc.

[1]short forms of words or phrases such as "4u" to represent "for you"



*Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 257–262,
Taipei, Taiwan, November 27 – December 1, 2017 

region. We propose two-level data-augmentation methods that do not use prior knowledge.

The contributions of this study are summarized as follows: (1) We propose two data-augmentation methods that generate synthetic data at character and morphological levels. (2) The experimental results with Japanese dialect text normalization demonstrate that our methods enable an encoder-decoder model, which performs poorly without data augmentation, to perform better than Moses, which is a strong baseline method when there is a small number of training examples.

## 2 Text Normalization using Encoder-Decoder Model

In this study, we focus on the dialect-normalization task as a text-normalization task. The input of this task is a dialect sentence, and the output is a "standard sentence" that corresponds to the given input dialect sentence. A "standard sentence" is written in normal form.

We model our dialect-normalization task by using a character-based attentional encoder-decoder model. More precisely, we use a single layer long short-term memory (LSTM) for both the encoder and decoder, where the encoder is bi-directional LSTM. Let $\boldsymbol{s} = (s_1, s_2, \ldots, s_n)$ be the character sequence of the (input) dialect sentence. Similarly, let $\boldsymbol{t} = (t_1, t_2, \ldots, t_m)$ be the character sequence of the (output) standard sentence. The notations $n$ and $m$ are the total lengths of the characters in $\boldsymbol{s}$ and $\boldsymbol{t}$, respectively. Then, the (normalization) probability of $\boldsymbol{t}$ given dialect sentence $\boldsymbol{s}$ can be written as

$$p(\boldsymbol{t}|\boldsymbol{s}, \theta) = \prod_{j=1}^{m} p(t_j|t_{<j}, \boldsymbol{s}), \quad (1)$$

where $\theta$ represents a set of all model parameters in the encoder-decoder model, which are determined by the parameter-estimation process of a standard softmax cross-entropy loss minimization using training data. Therefore, given $\theta$ and $\boldsymbol{s}$, our dialect normalization task is defined as finding $\boldsymbol{t}$ with maximum probability:

$$\hat{\boldsymbol{t}} = \mathrm{argmax}_{\boldsymbol{t}}\{p(\boldsymbol{t}|\boldsymbol{s}, \theta)\}, \quad (2)$$

where $\hat{\boldsymbol{t}}$ represents the solution.

## 3 Proposed Methods

This section describes our proposed methods for generating augmented-data. The goal with augmented data generation is to generate a large amount of corresponding standard and dialect sentence pairs, which are then used as additional training data of encoder-decoder models. To generate augmented data, we construct a model that converts standard sentences to dialect sentences since we can easily get a lot of standard sentences. Our methods are designed based on different perspectives, namely, morphological- and character-levels.

| standard morphs ($m_t$) | dialect morphs ($m_s$) | $p(m_s\|m_t)$ |
|---|---|---|
| し/ない (*shinai*) | せん (*sen*) | 0.767 |
| の/です/が (*nodesuga*) | ん/じゃ/けど (*njyakedo*) | 0.553 |
| ください (*kudasai*) | つか/あ/さい (*tukasai*) | 0.517 |

Table 1: Examples of extracted morphological conversion patterns

### 3.1 Generating Augmented Data using Morphological-level Conversion

Suppose we have a small set of standard and dialect sentence pairs and a large standard sentences. First, we extract morphological conversion patterns from a (small) set of standard and dialect sentence pairs. Second, we generate the augmented data using extracted conversion patterns.

**Extracting Morphological Conversion Patterns** For this step, both standard and dialect sentences, which are basically identical to the training data, are parsed using a pre-trained morphological analyzer. Then, the longest difference subsequences are extracted using dynamic programming (DP) matching. We also calculate the conditional generative probabilities $p(m_s|m_t)$ for all extracted morphological conversion patterns, where $m_s$ is a dialect morphological sequence and $m_t$ is a standard morphological sequence. We set $p(m_s|m_t) = F_{m_s,m_t}/F_{m_t}$, where $F_{m_s,m_t}$ is the joint frequency of $(m_s, m_t)$ and $F_{m_t}$ is the frequency of $m_t$ in the extracted morphological conversion patterns of training data. Table 1 gives examples of extracted patterns from Japanese standard and dialect sentence pairs, which we discuss in the experimental section.

**Generating Augmented Data using Extracted Morphological Conversion Patterns** After we obtain morphological conversion patterns, we generate a corresponding synthesized dialect sentence of each given standard sentence by using the ex-

**Algorithm 1** Generating Augmented Data using Morphological Conversion Patterns

```
morphlist ← MorphAnalyse(standardsent)
newmorphlist ← []
for i = 0 ... len(morphlist) do
    sent ← CONCAT(mrphlist[i:])
    MatchedList ← CommonPrefixSearch(PatternDict, sent)
    m_si ← SAMPLE(MatchedList, P(m_s|m_t))
    newmorphlist ← APPEND(newmorphlist, m_si)
end for
synthesizedsent ← CONCAT(newmrphlist)
return synthesizedsent
```

**input (standard sentence)**: インストールしなかった
morph sequence: インストール/し/なかっ/た/
matched conversion pattern:
$(m_t, m_s)$ = (し/なかっ/た, せん/かった)
replaced morph sequence: インストール/せん/かった
**output (augmented sentence)**: インストールせんかった

Table 2: Example of generated augmentation data using morphological conversion patterns

tracted morphological conversion patterns. Algorithm 1 shows the detailed procedure of generating augmented data.

More precisely, we first analyze the standard sentences with the morphological analyzer. We then look up the extracted patterns for the segmented corpus from left to right and replace the characters according to probability $p(m_s|m_t)$. Table 2 shows an example of generated augmentation data. When we sample dialect pattern $m_s$ from MatchedList, we use two types of $p(m_s|m_t)$. The first type is fixed probability. We set $p(m_s|m_t)$ = 1/len(MatchedList) for all matched patterns. The second type is generative probability, which is calculated from the training data (see the previous subsection). The comparison of these two types of probabilities is discussed in the experimental section.

### 3.2 Generating Augmented Data using Character-level Conversion

For our character-level method, we take advantage of the phrase-based SMT toolkit Moses for generating augmented data. The idea is simple and straightforward; we train a 'standard-to-dialect' sentence SMT model at a character-level and apply it to a large non-annotated standard sentences. This model converts the sentence by using character phrase units. Thus, we call this method "character-level conversion".

### 3.3 Training Procedure

We use the following two-step training procedure. (1) We train model parameters by using both human-annotated and augmented data. (2) We then retrain the model parameters only with the human-annotated data, while the model parameters obtained in the first step are used as initial values of the model parameters in this (second) step. We refer to these first and second steps as "pre-training" and "fine-tuning", respectively. Obviously, the augmented data are less reliable than human-annotated data. Thus, we can expect to improve the performance of the normalization model by ignoring the less reliable augmented data in the last-mile decision of model parameter training.

## 4 Experiments

### 4.1 Data

The dialect data we used were crowdsourced data. We first prepared the standard seed sentences, and crowd workers (dialect natives) rewrote the seed sentences as dialects. The target areas of the dialects were Nagoya (NAG), Hiroshima (HIR), and Sendai (SEN), which are major cities in Japan. Each region's data consists of 22,020 sentence pairs, and we randomly split the data into training (80%), development (10%), and test (10%). For augmented data, we used the data of Yahoo chiebukuro, which contains community QA data. Since the human-annotated data are spoken language text, we used the community QA data as close-domain data.

### 4.2 Settings

For the baseline model other than encoder-decoder models, we used Moses. Moses is a tool of training statistical machine translation and a strong baseline for the text-normalization task (Junczys-Dowmunt and Grundkiewicz, 2016). For such a task, we can ignore the word reordering; therefore, we set the distortion limit to 0. We used MERT on the development set for tuning. We confirmed that using both manually annotated and augmented data for building LM greatly degraded its final BLUE score in our preliminary experiments and used only manually annotated data as the training data of LM.

We used beam search for the encoder-decoder model (EncDec) and set the beam size to 10. When in the n beam search step, we used length normalized score $S(t, s)$, where

| method | BLEU | | |
|---|---|---|---|
| | NAG | HIR | SEN |
| No-transformation | 72.4 | 63.9 | 57.3 |
| Moses (train) | **80.1** | 72.3 | 67.1 |
| Moses (train + mr:R) | 75.4 | 71.0 | 64.9 |
| Moses (train + mr:W) | 80.0 | 73.7 | 67.7 |
| Moses (train + mo) | 79.9 | 74.3 | 66.9 |
| Moses (train + mo + mr:W) | 80.0 | 73.3 | 67.8 |
| EncDec (train) | 43.3 | 33.9 | 27.6 |
| EncDec (train + mr:R) | 75.3 / 63.5 | 69.0 / 67.3 | 64.2 / 58.8 |
| EncDec (train + mr:W) | 78.6 / 78.2 | 74.9 / 73.5 | 68.0 / 67.6 |
| EncDec (train + mo) | 79.1 / 79.1 | 74.2 / 72.9 | 66.9 / 65.6 |
| EncDec (train + mo+mr:W) | **80.1** / 79.5 | **75.5** / 74.6 | **68.2** / 68.1 |

Table 3: BLEU scores of normalization. "/" indicates with (left) and without (right) fine tuning. 200,000 pairs of augmented data were used.

| method | BLEU | | |
|---|---|---|---|
| | NAG | HIR | SEN |
| Moses (oracle) | 80.2 | 75.7 | 68.3 |
| Moses (best) | 80.1 | 74.3 | 67.8 |
| EncDec (oracle) | 84.8 | 81.6 | 73.1 |
| EncDec (best) | 80.1 | 75.5 | 68.2 |

Table 4: Evaluation of oracle sentences

$S(t, s)$ =log$(p(\boldsymbol{t}|\boldsymbol{s}, \theta))/|\boldsymbol{t}|$. We maximize $S(t, s)$ to find normalized sentence. We set the embedding size of the character and hidden layer to 300 and 256, respectively. We used "mrphaug (mr)" as the augmented data generated from morphological-level conversion and "mosesaug (mo)" as augmented data generated from character-level conversion (Moses). The "mr:R" and "mr:W" represent the difference in generative probability $p(m_s|m_t)$, which is used when generating augmented data; "mr:R" indicates fixed generative probability and "mr:W" indicates weighted generative probability. For the evaluation, we used BLEU (Papineni et al., 2002), which is widely used for machine translation.

### 4.3 Results

Table 3 lists the normalization results. No-transformation indicates the result of evaluating input sentences without transformation. Moses achieved a reasonable BLEU score with a small amount of human-annotated data. However, the improvement of adding augmented data was limited. On the other hand, the encoder-decoder model showed a very low BLEU score with a small amount of human-annotated data. With this amount of data, the encoder-decoder model generated a sentence that was quite different from the reference. When adding augmented data, the BLEU score improved, and fine tuning was effective for all cases.

When comparing our augmented-data-generation methods, generating data according to fixed probability (mr:R) degraded the BLEU score both for Moses and the encoder-decoder model. When generating data with fixed probability, the quality of augmented data becomes quite low. However, by generating data according to generative probability (mr:W), which is estimated with training data, the BLEU score improved. This indicates that when generating data using morphological-level Conversion, it is important to take into account the generative probability. Combining "mr:W" and "mo" (train+mo+mr:W) achieves higher BLEU scores than that of other methods. This suggests that combining different types of data will have a positive effect on normalization accuracy.

When comparing three difference regions, the BLUE scores of Moses (train) and EncDec (train+mo+mr:W) for NAG (Nagoya) were the same score, while there were improvements for HIR (Hiroshima) and SEN (Sendai). It is inferred that the effect of the proposed methods for NAG were limited because the difference between input (dialect) sentences and correct (standard) sentences was small.

## 5 Discussion

**Oracle Analysis** To investigate the further improvement on normalization accuracy, we analyzed oracle performance. We enumerated the

top 10 candidates of normalized sentences from Moses and proposed method, extracted the candidates that were the most similar to the reference, and calculated the BLEU scores. Table 4 shows the results of oracle performance. Interestingly, the oracle performances of the encoder-decoder model with augmented data was quite high, while that of Moses was almost the same as the best score. This implies that there is room for improvement for the encoder-decoder model by just improving the decoding or ranking function.

**Other text normalization task** In this study, we evaluated our methods with Japanese dialect data. However, these methods are not limited to Japanese dialects because they do not use dialog-specific information. If there is prior knowledge, the combination of them will be more promising for improve normalization performance. We will investigate the effectiveness of our methods for other normalization tasks for future work.

**Limitation** Since our data-augmentation methods are based on human-annotated training data, the variations in the generated data depend on the amount of training data. The variations in augmented data generated with our data-augmentation methods are strictly limited within those appearing in the human-annotated training data. This essentially means that the quality of augmented data deeply relies on the amount of (human-annotated) training data. We plan to develop more general methods that do not deeply depend on the amount of training data.

## 6 Conclusion

We investigated the effectiveness of our augmented-data-generation methods for neural text normalization. From the experiments, the quality of augmented data greatly affected the BLEU score. Moreover, a two-step training strategy and fine tuning with human-annotated data improved this score. From these results, there is possibility to improve the accuracy of normalization if we can generate higher quality data. For future work, we will explore a more advanced method for generating augmented data.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.*, pages 5:1–5:27.

Fei Huang. 2015. Improved arabic dialect classification with social media data. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2118–2126, Lisbon, Portugal. Association for Computational Linguistics.

Taishi Ikeda, Hiroyuki Shindo, and Yuji Matsumoto. 2017. Japanese text normalization with encoder-decoder model. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on EMNLP*.

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 on Association for Computational Linguistics*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal arabic to english. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Sofia, Bulgaria. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Ziang Xie, Anand Avati, Naveen Arivazhagan, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*.

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*.