Automatic Error Correction on Japanese Functional Expressions Using Character-based Neural Machine Translation

Jun Liu[1], Fei Cheng[2], Yiran Wang[1], Hiroyuki Shindo[1,3], Yuji Matsumoto[1,3]
[1] Nara Institute of Science and Technology, Japan
[2] National Institute of Informatics, Japan
[3] RIKEN Center for Advanced Intelligence Project (AIP), Japan
{liu.jun.lc3, wang.yiran.ws5, shindo, matsu}@is.naist.jp
fei-cheng@nii.ac.jp

Abstract

Correcting spelling and grammatical errors of Japanese functional expressions shows practical usefulness for Japanese Second Language (JSL) learners. However, the collection of these types of error data is difficult because it relies on detecting Japanese functional expressions first. In this paper, we propose a framework to correct the spelling and grammatical errors of Japanese functional expressions as well as the error data collection problem. Firstly, we apply a bidirectional Long Short-Term Memory with a Conditional Random Field (BiLSTM-CRF) model to detect Japanese functional expressions. Secondly, we extract phrases which include Japanese functional expressions as well as their neighboring words from native Japanese and learners' corpora. Then we generate a large scale of artificial error data via substitution, injection and deletion operations. Finally, we utilize the generated artificial error data to train a sequence-to-sequence neural machine translation model for correcting Japanese functional expression errors. We also compare the character-based method with the word-based method. The experimental results indicate that the character-based method outperforms the word-based method both on artificial error data and real error data.

1 Introduction

The Japanese Language has various types of functional expressions which consist of more than one word including both content words and functional words, such as "を踏まえて (based on), に違いない (no doubt), てはいけない (must not)". Due to the various meanings and usages, spelling and grammatical errors are often made by JSL learners when they use Japanese functional expressions in their writings. We observed some example sentences in Lang-8 Learner Corpora[1] and summarized some typical types of spelling and grammatical errors of Japanese functional expressions, including word selection error (S), missing word error (M), redundant error (R), and word spelling error (W). Some example sentences of grammatical errors are shown in Table 1. Much previous research has paid special attention to the automatic detection of Japanese functional expressions (Tsuchiya et al., 2006; Shime et al., 2007; Suzuki et al., 2012) while relatively few grammatical error correction applications have been developed to support JSL learners. Given this situation, automatic grammatical error correction of sentences written by JSL learners is essential in Japanese language learning.

In this paper, we define a new task of correcting spelling and grammatical errors on Japanese functional expressions as follows. Given a phrase of a Japanese functional expressions and its neighboring words, our system aims to correct errors inside this phrase. For instance, a phrase " 行くましょう。 (Let's go.)" where the Japanese functional expression is in bold will be expected to be corrected as "行きましょう。", because the correct usage of Japanese verb conjugation rules in this phrase depends on the Japa-

[1] http://lang-8.com

nese functional expression “ましょう (Let’s)”. However, collecting a large number of available real error phrases written by JSL language learners is not easy because of relying on detecting Japanese functional expressions first. To solve this problem, we first detect the Japanese functional expressions using a BiLSTM-CRF model. Next, we extract phrases including Japanese functional expressions as well as their neighboring words for generating artificial error data. For automatic error correction, we utilize a neural sequence-to-sequence model to treat spelling and grammatical error correction as a translation process from incorrect character sequences to correct character sequences. We also conduct our experiments with the word-based method as a baseline for comparison.

Error Type	Example Sentences
word selection error (S)	Incorrect Sentence: 綺麗な日本に行く**ましょう**。 Correct Sentence: 綺麗な日本に行き**ましょう**。 (Let’s go to the beautiful Japan.)
missing word error (M)	Incorrect Sentence: このドラマのかげで、やる気をもらいました。 Correct Sentence: このドラマ**のおかげで**、やる気をもらいました。 (I got motivated because of this drama.)
redundant error (R)	Incorrect Sentence: 辞職したの後で新しい会社で働きました。 Correct Sentence: 辞職し**た後で**新しい会社で働きました。 (I worked in a new company after retirement.)
word spelling error (W)	Incorrect Sentence: 私にもできるかもしらない。 Correct Sentence: 私にもできる**かもしれない**。 (Maybe I also can do it.)

Table1: Typical examples of grammatical errors of Japanese functional expressions.
In the sentences, Japanese functional expressions are in bold, while errors are underlined.

The remainder of this paper is organized as follows: Section 2 reviews some related work on spelling and grammatical error correction. Section 3 introduces language resources used in our work for training BiLSTM-CRF model and generating artificial error data. Section 4 describes how BiLSTM-CRF model is used to detect Japanese functional expressions and Section 5 explains the method for generating artificial error data. In Section 6, we conduct the experiments using neural machine translation and analyze the results. Section 7 concludes with a summation of this work and describes our future work.

2 Related Work

Spelling correction is an automatic algorithm for detecting and correcting human spelling errors in every written language, which has been an active research in Natural Language Processing (NLP) (Sun et al. 2010; Chen et al. 2013; Liu et al. 2013; Liu et al. 2015;).

Grammatical error correction (GEC) is a task of detecting and correcting grammatical errors in text written by native language writers or non-native foreign language writers. over the past few decades, GEC in English has been widely researched, such as Helping Our Own (Dale and Kigarriff, 2011; Dale et al., 2012), CoNLL Shared Task (Ng et al., 2013; Ng et al., 2014). Many shared tasks on GEC for Chinese Second Language Learners have also been held, such as the NLP-TEA Shared Task (Yu et al., 2014; Lee et al., 2015; Lee et al., 2016; Rao et al., 2017; Rao et al., 2018). On Japanese GEC, much work has been done on particle error correction for JSL learners (Oyama, 2010; Ohki et al., 2011; Mizumoto et al., 2011; Imamura et al., 2014).

Collecting large-scale annotated error data written by second language learners is not so easy. To cope with grammatical error data scarcity, several studies proposed effective approaches for generating artificial error data (Irmawati et al., 2017; Rei et al., 2017).

Several approaches of using Statistical machine translation (SMT) for GEC have been proposed (Brockett et al., 2006; Mizumoto et al., 2011; Mizumoto et al., 2015). Recently, neural networks have shown success in many NLP tasks, such as machine translation (MT) (Eriguchi et al., 2016; Gehring et al., 2017), named entity recognition (NER) (Kuru et al., 2016; Misawa et al., 2017) and etc. For GEC, several studies have applied neural machine translation (NMT) approach (Chollampatt et al., 2016; Yuan and Briscoe, 2016). NMT is applied in the GEC task as it may be possible to correct erroneous phrases and sentences that have not been seen in the training data more effectively (Luong et al., 2015). NMT-based systems thus may help ameliorate the shortage of large error-annotated learner corpora for GEC.

As previous research mentioned above, few studies have aimed at spelling and grammatical error corrections on Japanese functional expressions. Therefore, our paper is an attempt to do this work using neural machine translation.

3 Language resources

We use the following corpora for training the BiLSTM-CRF model to detect Japanese functional expressions. We use Lang-8 Learner, Tatoeba, HiraganaTimes corpora for generating artificial error data, because these three corpora are particularly designed for Japanese second language learners, in which the sentences are easy for them to read and understand. The details of these corpora are as follows.

・**Lang-8 Learner Corpora**: this is a large-scale error-annotated learner corpora, covering 80 languages. We use only the Lang-8 corpus of Japanese learners, which consists of approximately 2M sentences.

・**Tatoeba**[2]: this corpus is a free online database of example sentences written by foreign language learners. We use only Japanese sentences (approximately 170K) from this corpus.

・**HiraganaTimes**[3]: this corpus is a Japanese-English bilingual corpus of magazines articles, which introduces Japan to non-Japanese, covering a wide range of topics including society, culture, history, etc. We use only Japanese sentences (approximately 150K) from this corpus.

・**BCCWJ**[4]: The Balanced Corpus of Contemporary Written Japanese (BCCWJ) is a corpus created for comprehending the breadth of contemporary written Japanese. The data comprises 104.3 million words, covering genres including general books and magazines, newspapers, business reports, blogs, internet forums, textbooks, and legal documents.

4 Detection of Japanese Functional Expressions

4.1 Data Pre-processing

Since we treat the Japanese functional expressions detection task as a character-based sequence labeling problem, we split the word in a sentence to character level by attaching position labels from a tag set: {B, I, E, O, B-SP, I-SP, E-SP, O-SP}. Here, we have tag 'B' indicating the beginning position of a word, 'I' indicating the middle position of a word, 'E' indicating the end position of a word, 'O' indicating a single character word, 'BSP' indicating the beginning position of a Japanese functional expression, 'I-SP' indicating the middle position of a Japanese functional expression, 'E-SP' indicating the end position of a Japanese functional expression, 'O-SP' indicating a Japanese functional expression with a single character word. Table 2 shows an example sentence (アメリカへ行きましょう。"Let's go to America.") after pre-processing.

Character	Label
ア	B
メ	I
リ	I
カ	E
へ	O
行	B
き	E
ま	B-SP
し	I-SP
ょ	I-SP
う	E-SP
。	O

Table 2: An example sentence after pre-processing

4.2 BiLSTM-CRF Model

The BiLSTM-CRF model (Huang et al., 2015) consists of three major parts: the embedding layer, the bi-directional LSTM layer, and the CRF layer.

As shown in figure 1, every character in sentence is represented as character embedding as input. The bidirectional LSTM layer is used to operate sequential information in two opposite directions. The CRF layer predicts correlated tag sequence under consideration of outputs from the

[2] https://tatoeba.org/eng/

[3] http://www.hiraganatimes.com/

[4] http://pj.ninjal.ac.jp/corpus_center/bccwj/en/

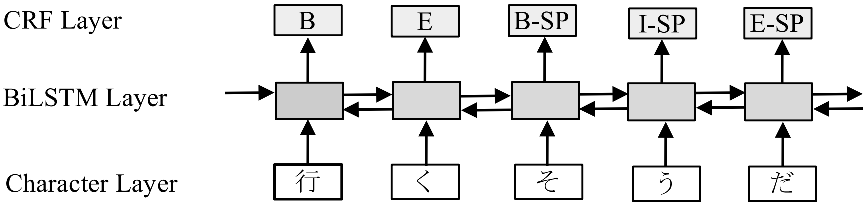

Figure 1: Structure of BiLSTM-CRF model

No.	Example Sentence	Result
1	Input: 雨が降っている。 (It is raining.) Output: 雨 が 降っ **ている** 。	Correct
2	Input: 静かにしてください。 (Please be quiet.) Output: 静か に し **てください** 。	Correct
3	Input: 彼女は重い病気にかかっている。 (She is suffering from a serious disease.) Output: 彼女 は 重い 病気 **にかかっている** 。 Gold result: 彼女 は 重い 病気 に かかっ **ている** 。	Incorrect

Table 3: Examples of detection of Japanese functional expressions.
In the sentences, Japanese functional expressions are in bold.

LSTM layer. In this work, we adopt a BiLSTM-CRF implementation[5] with the following empirical hyper-parameter setting. We apply 300-dimensional randomly initialized character embddings and 300-dimensional hidden state for LSTM. We choose Adam as the optimizer and set the learning rate equal to 0.001.

4.3 Experiment and Evaluation

We collect some sentences containing Japanese functional expressions from the following corpora: Lang-8 Learner, Tatoeba, HiraganaTimes, BCCWJ. In addition, we also collect sentences from some Japanese functional expression dictionaries (Group Jamashi and Xu, 2001; Xu and Reika, 2013).

As the results, we use 21,458 sentences for training data, and 916 sentences for test data. In the training data and test data, some sentences collected from Lang-8 Learner Corpora contain real spelling errors, since we would like to see if the Bi-LSTM-CRF model can detect Japanese functional expressions with spelling errors. All the sentences are first segmented into individual words using a free Japanese morphological analyzer MeCab[6]. Then the words are split into characters and manually annotated with tags after pre-processing.

As evaluation metrics, we use precision, recall and F_1-score as shown in the following formulas. We evaluate the output of Japanese functional expressions as a whole word level. Table 3 demonstrates certain examples of Japanese functional expressions. For example, Japanese functional expressions in sentences No.1 and No.2 were correctly identified, while the system wrongly identified content words as a Japanese functional expression in sentence No.3. The final experimental result is shown in Table 4.

$$Precision = \frac{correctly\ identified\ results\ of\ system}{identified\ results\ of\ system}$$

$$Recall = \frac{correctly\ identified\ results\ of\ system}{results\ in\ test\ data}$$

$$F_1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Precision	**Recall**	**F_1-score**
88.38%	90.34%	89.35%

Table 4: Experimental result of Detection of Japanese functional expressions

5 Artificial Error Generation

In this section, we apply our Japanese functional expression detector, which is trained with the BiLSTM-CRF model in Section 4.2 to extract phrases which include Japanese functional expressions with their neighboring words for generating artificial error data. Our method mainly consists of two steps, as shown in Figure 2.

[5] https://github.com/Determined22/zh-NER-TF
[6] http://taku910.github.io/mecab/

In Step 1, we first extracted real error phrases from Lang-8 Leaner Corpora using the BiLSTM-CRF model. As the results, we extracted total 609 real error phrases. According to our observation, every real error phrase contains only one grammatical error or one spelling error on Japanese functional expression. Since the data of real error phrases is very small, which is not far from enough for training data, we then extracted phrases in corrected sentences from Lang-8 Leaner Corpora, and native phrases from Tatoeba and HiraganaTimes corpora. Table 5 shows several extraction results of phrases of Japanese functional expressions.

In Step 2, we randomly selected 309 real error phrases extracted in Step 1 as the error templates and the remaining 300 real error phrases were used as test data in our error correction task. We generated artificial error data by using the following three operations to imitate typical errors: Substitution, Injection and Deletion. In particular, we generated artificial error data by imitating the error templates when using injection and deletion operations accounted for the majority. Table 6 shows a few examples of artificial error generation. As the results, we generated 396,663 phrase pairs of artificial error data. The same as the real error phrase, every artificial error phrase also involves only one grammatical error or one spelling error on Japanese functional expression.

・**Substitution**:

This method replaces a correct verb that appear just before a Japanese functional expression with its other conjugated forms.

・**Injection**:

This method injects a redundant character in a Japanese functional expression or in its neighboring word.

・**Deletion**:

This method deletes a character in a Japanese functional expression or in its neighboring word.

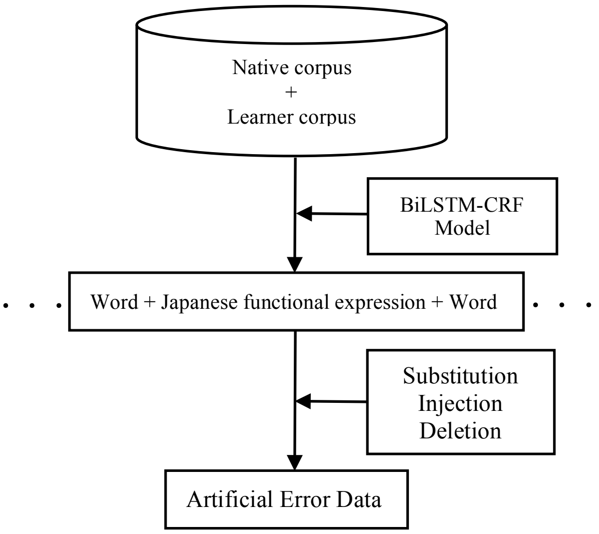

Figure 2: The steps in artificial error generation

Input: あなたは薬を飲ま**なければならない**。 (You must take the medicine.) Output: あなた は 薬 を 飲ま **なければならない** 。 Extracted phrase: 飲ま **なければならない** 。
Input: ラジオを修理する**ために**分解した。 (I took the radio apart to repair it.) Output: ラジオ を 修理 する **ために** 分解 し た 。 Extracted phrase: する **ために** 分解
Input: 彼は天才**かもしれない**。 (He may be a genius.) Output: 彼 は 天才 **かもしれない** 。 Extracted phrase: 天才 **かもしれない** 。

Table 5: Extraction results of phrases of Japanese functional expressions.
In the sentences, Japanese functional expressions are in bold.

Method	Example
Substitution	Extracted phrase: 飲ま **なければならない**。 Artificial error data: 飲む **なければならない**。 飲み **なければならない**。 飲ん **なければならない**。 飲も **なければならない**。
Injection	Extracted phrase: する **ために** 分解 Artificial error data: する の**ために** 分解
Deletion	Extracted phrase: 多い **おかげで** 彼 Artificial error data: 多い かげで 彼

Table 6: Examples of artificial error generation.
In the sentences, Japanese functional expressions are in bold, while artificial errors are underlined.

6 Automatic Error Correction

6.1 Neural Sequence-to-Sequence Model

In this paper, spelling and grammatical error correction is treated as a translation task from incorrect phrases into correct phrases. Based on empirical observation, correcting grammatical errors on Japanese functional expressions can be mainly seen as substitution, injection, deletion operations of characters. The character-based translation process is a natural choice to handle this task. In the meanwhile, the word-based process will suffer from the sparsity of error types, especially when facing the real data. Therefore, we proposed a character-based neural sequence-to-sequence model for the task of correcting grammatical errors on Japanese functional expressions. We also perform the word-based process as a baseline for comparison.

The neural sequence-to-sequence model, consists of two main pieces: an encoder that processes the input and a decoder that generates the output. Both the encoder and the decoder are recurrent neural network (RNN) layers that can be implemented using a vanilla RNN, a Long Short-term Memory (LSTM), or a gated recurrent unit (GRU). In the basic sequence-to-sequence model, the encoder processes the input sequence into a fixed representation that is fed into the decoder as a context. The decoder then uses some mechanism to decode the processed information into an output sequence. The basic architecture is shown in Figure 3 (Sutskever et al., 2014; Cho et al., 2014). In this paper, we trained a 2-layer LSTM sequence-to-sequence model with 128-dim hidden units and embeddings for 12 epochs. We used a drop value of 0.2. The formulas of LSTM can be found in the following equations, where the $W_{i,f,g,o}^{(l)}$, $U_{i,f,g,o}^{(l)}$ and $b_{i,f,g,o}^{(l)}$ are the l-th layer's trainable parameters, the $\odot$ means point-wise multiplication and the σ and tanh refers to sigmoid and hyperbolic tangent function respectively. The hidden state of current layer $h_t^{(l)}$ will be fed to next layer as input $x_t^{(l+1)}$.

$$
\begin{aligned}
i_t^{(l)} &= \sigma\left(W_t^{(l)} x_t^{(l)} + U_t^{(l)} h_{t-1}^{(l)} + b_i^{(l)}\right) \\
f_t^{(l)} &= \sigma\left(W_t^{(l)} x_t^{(l)} + U_t^{(l)} h_{t-1}^{(l)} + b_f^{(l)}\right) \\
g_t^{(l)} &= \tanh\left(W_t^{(l)} x_t^{(l)} + U_t^{(l)} h_{t-1}^{(l)} + b_g^{(l)}\right) \\
o_t^{(l)} &= \sigma\left(W_t^{(l)} x_t^{(l)} + U_t^{(l)} h_{t-1}^{(l)} + b_o^{(l)}\right) \\
c_t^{(l)} &= f_t^{(l)} \odot c_{t-1}^{(l)} + i_t^{(l)} \odot g_t^{(l)} \\
h_t^{(l)} &= o_t^{(l)} \odot \tanh c_t^{(l)}
\end{aligned}
$$

6.2 Experimental Settings

As mentioned in Section 5, we ultimately got 396,663 artificial error phrase pairs. In the first experiment, we used 326,663 phrase pairs for training data, 35,000 phrase pairs for development data, and 35,000 phrase pairs for test data. In the final experiment, we used the remaining 300 real error phrase pairs mentioned in Section 5 for another test data.

In both experiments, we proposed two methods: one is the word-based method where the input phrase is split to word sequences, the other is character-based method where the input phrase is split to character sequences. We performed the word-based method as a baseline for comparison.

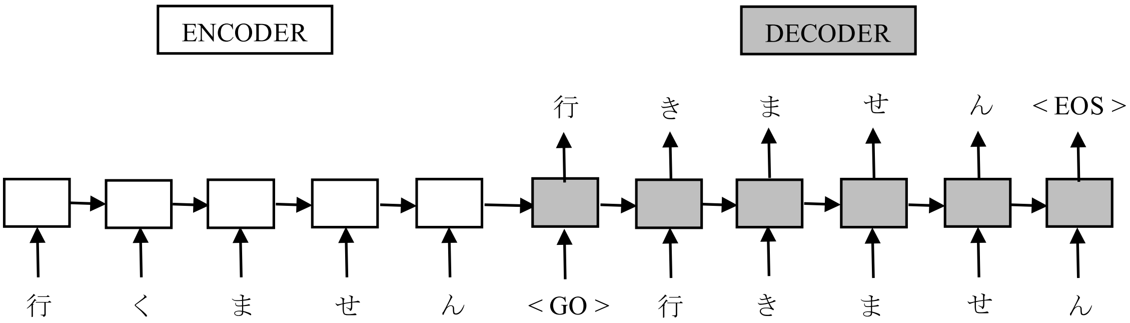

Figure 3: The basic architecture of sequence-to-sequence model

Data	Method	Precision	Recall	F_1-score
Artificial error	word-based	71.86%	71.74%	71.80%
	character-based	94.20%	93.98%	94.09%
Real error	word-based	63.88%	63.67%	63.77%
	character-based	92.25%	87.33%	89.72%

Table 7: Experimental results of error correction on Japanese functional expressions.

No.	Word-based		Character-based
1	input	助ける **ましょう** ！	助ける**ましょう**！
	output	閉じ **ましょう** ！(Wrong)	助け**ましょう**！(Correct)
2	input	員 のかげで 、	員のかげで、
	output	呼びかけ **のおかげで** 、(Wrong)	員**のおかげで**、(Correct)
3	input	飾る **のために** 、	飾る**のために**、
	output	助ける **ために** 、(Wrong)	飾る**ために**、(Correct)
4	input	思い浮かぶ かもしりません 。	思い浮かぶかもしりません。
	output	近い **かもしれません** 。(Wrong)	思い浮かぶ**かもしれません**。(Correct)
5	input	直せ **ないといけません** ね	直せ**ないといけません**ね
	output	直せ **ないといけません** ね (Wrong)	直さ**ないといけません**ね (Correct)
6	input	する **後に** 挨	する**後に**挨
	output	し **た後に** 眠く (Wrong)	し**た後に**資 (Wrong)
	gold result	し **た後に** 挨	し**た後に**挨
7	input	探せ **ないといけません** 。	探せ**ないといけません**。
	output	とら **ないといけません** 。(Wrong)	探せ**ないといけません**。(Wrong)
	gold result	探さ **ないといけません** 。	探さ**ないといけません**。
8	input	それの **ために** 流行	それの**ために**流行
	output	それの **ために** 通り過ぎ (Wrong)	それ**ために**行眠 (Wrong)
	gold result	その **ために** 流行	そ の**ために**流行

Table 8: Examples of system outputs tested on real error data.
In the phrases, the Japanese functional expressions are in bold, while errors are underlined.

6.3 Experimental Results

In this section, we evaluate our error correction model in both the artificial data and the real data. As we described in Section 5, the generation of the artificial data is based on 309 error templates. It suggests that the error types in the artificial test data are relatively more overlapped to the training data, compared to the real situation. For this reason, we perform the experiment with 300 real error data, which contain more unseen error types. The results can fairly reflect the generalization ability of our model.

As evaluation metrics, we use precision, recall and F_1-score based on words and characters. Table 7 shows the final experimental results of grammatical error correction tested both on artificial error data and real error data. According to the results, the character-based method achieved much higher F-score than the word-based method both on artificial error data and real error data, indicating that the character-based neural sequence-to-sequence model is more effective than the word-based neural sequence-to-sequence model. When using the character-based method, we also got a higher F-score tested on the artificial error data than real error data. As expected, the real test data results are lower than the artificial test data. The real test data contains more unknown error types, which provides a more practical and meaningful evaluation.

6.4 Error analysis

Some examples of system results tested on real data are shown in Table 8.

On primary cause of deterioration of F_1-score using the word-based method is that the system wrongly corrected the neighboring words into other words, such as examples 1-4 and examples 6-8 in Table 8, although the system was able to correct Japanese functional expressions. Similarly, the errors occurred when using the character-based method, such as examples 6 and 8 in Table8.

Additionally, the failure of detecting grammatical errors also caused errors, such as the example 5 when using the word-based method and example 7 when using the character-based method.

7 Conclusions and Future Work

In this paper, we define a new task of correcting spelling and grammatical errors on Japanese functional expressions. Our BiLSTM-CRF model can precisely recognize Japanese functional expressions and their neighboring words as the correction targets. Considering the real error data is insufficient, we generated artificial error data via substitution, injection, deletion of characters in correct data. To do error correction, we utilized neural machine translation, to train a word-based sequence-to-sequence model and a character-based sequence-to-sequence model, respectively. Experimental results indicated that the character-based method achieved much higher F-score than the word-based method.

In the future, we plan to extract more neighboring words of Japanese functional expressions to correct more errors, especially Japanese functional expressions with two or more meanings and usages, which we did not handle in this paper. Moreover, we want to apply the artificial error data to generate multiple-choice questions for JSL learners in a Japanese functional expression learning system.

Acknowledgments

We would like to thank the reviewers for their valuable comments and suggestions that further improved our work.

References

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 249-256.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang and Hsin-Hsi Chen. 2013. A study of language modeling for Chinese spelling check. In *Proceedings of the 6th International Conference on Natural Language Processing*.

Kyunghyum Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724-1734.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural Network Translation Models for Grammatical Error Correction. In *Proceeding of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 2768-2774.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*, pages 242-249.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 54-62.

Akiko Eriguchi, Kazuma Hashimoto and Yoshisama Tsuruoka. 2016. Tree-to-Sequence Attentional Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'2016),* pages 823-833.

Jonas Gehring, Michael Auli, David Grangier and Yann N. Dauphin. 2017. A Convolutional Encoder Model for Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'2017),* pages 123-125.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991.*

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu and Hitoshi Nishikawa. 2014. Particle Error Correction from Small Error Data for Japanese Learners. *Journal of Natural Language Processing 21(4)*, pages 941-963.

Budi Irmawati, Hiroyuki Shindo and Yuji Matsumoto. 2017. Generating Artificial Error Data for Indonesian Preposition Error Corrections. *International Journal of Technology (2017)3*, pages:549-558.

Group Jamashi and Yiping Xu. 2001. Chubunban Nihongo Kukei Jiten-Nihongo Bunkei Jiten (in Chinese and Japanese). *Tokyo: Kurosio Publishers*.

Onur Kuru, Arkan Ozan Can, and Deniz Yuret. 2016. Charner: Character-level named entity recognition. In *proceeding of the 26th International Conference on Computational Linguistics*, pages 911-921.

Lung-Hao Lee, Gaoqi Rao, Liang-Chih Yu, Xun Endong Xun, Baolin Zhang, and Li-Ping Chang. 2016. Overview of the NLP-TEA 2016 Shared Task for Chinese Grammatical Error Diagnosis. In *Proceedings of the Workshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA'16)*, pages1-6.

Lung-Hao Lee, Liang-Chih Yu, and Li-Ping Chang. 2015. Overview of the NLP-TEA2015 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniquesfor Educational Applications (NLP-TEA'15)*, pages 1-6.

Xiaodong Liu, Fei Cheng, Kevin Duh and Yuji Matsumoto. 2015. A Hybrid Ranking Approach to Chinese Spelling Check. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), Vol 14, No. 14, Article 16.*

Xiaodong Liu, Fei Cheng, Yanyan Luo, Kevin Duh and Yuji Matsumoto. 2013. A hybrid Chinese spelling correction system using language model and statistical machine translation with remarking. In *Proceedings of the 6th International Joint Conference on Natural Language Processing,* pages 54.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the Rare Word Problem in Neural Machine Translation. In *Proceedings of the ACL-IJCNLP*, pages 11-19.

Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura and Tomoko Ohkuma. 2017. Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition. In *proceeding of the First Workshop on Subword and Character Level Models in NLP*, pages 97-102.

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 the 5th International Joint Conference on Natural Language Processing*, pages 147-155.

Tomoya Mizumoto; Masato Mita; Yuji Matsumoto. 2015. Grammatical Error Correction Considering Multi-Word Expressions. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*. pages 82–86.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task*, pages 1-14.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1-12.

Megumi Ohki, Hiromi Oyama., Kei Kitauchi, Takashi Suenaga, and Yuji Matsumoto. 2011. Judgment of Incorrect Usage of Particles in Documents for System Development Written by Non-native Japanese Speakers. In *Proceedings of the 17th Annual Meeting of the Association for Natural Language Processing*, pages 1047-1050.

Horomi Oyama. 2010. Automatic Error Detection Method for Japanese Particles. *Polyglossia* Vol.1,55-63, pages 55-63.

Gaoqi Rao, Qi Gong, Baolin Zhang, Endong Xun. 2018. Overview of NLPTEA-2018 Share Task Chinese Grammatical Error Diagnosis. In *Proceeding of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42-51.

Gaoqi Rao, Baolin Zhang, Endong Xun, Lung-Hao Lee. 2017. Chinese Grammatical Error Diagnosis. In *Proceedings of the Workshop on NLP Techniques for Educational Applications, The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017) Shared tasks*, pages 1-8.

Marek Rei, Mariano Felice, Zheng Yuan, Ted Briscoe. 2017. Artificial Error Generation with Machine Translation and Syntactic Patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287-292.

Takao Shime, Masatoshi Tsuchiya, Suguru Matsuyoshi, Takehito Utsuro, Satoshi Sato. 2007. Automatic Detection of Japanese Compound Functional Expressions and its Application to Statistical Dependency Analysis, *Journal of Natural Language Processing*, Vol (14). No.5, pages 167-196.

Xu Sun, Jianfeng Gao, Daniel Micol and Chris Quirk. 2010. Learning phrase-based spelling error models from click through data. In *proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10)*, pages 266-274.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. *In Advances in Neural Information Processing Systems*, pages 3104-3112.

Takafumi Suzuki, Yusuke Abe, Itsuki Toyota, Takehito Utsuro, Suguru Matsuyoshi, Masatoshi Tsuchiya. 2012. Detecting Japanese Compound Functional Expressions using Canonical/Derivational Relation, In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*.

Masatoshi Tsuchiya, Takao Shime, Toshihiro Takagi, Takehito Utsuro, Kiyotaka Uchimoto, Suguru Matsuyoshi, Satoshi Sato, Seiichi Nakagawa. 2006. Chunking Japanese compound functional expressions by machine learning, In *Proceedings of the Workshop on Multi-word-expressions in a Multilingual Context*. pages 25-32.

Xiaoming Xu and Reika. 2013. Detailed introduction of the New JLPT N1-N5 grammar. *East China University of Science and Technology Press*.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning Chinese as foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications (NLP-TEA'14)*, pages 42-47.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceeding of NAACL-HLT, 2016*. pages 380-38.