Grammatical-Error-Aware Incorrect Example Retrieval System for Learners of Japanese as a Second Language

Mio Arai **Masahiro Kaneko** **Mamoru Komachi**
Tokyo Metropolitan University
Hino City, Tokyo, Japan
{arai-mio@ed., kaneko-masahiro@ed., komachi@}tmu.ac.jp

Abstract

Existing example retrieval systems do not include grammatically incorrect examples, or only present a few examples, if any. Even if a retrieval system has a wide coverage of incorrect examples along with the correct counterparts, learners need to know whether their query includes errors. Considering the usability of retrieving incorrect examples, our proposed method uses a large-scale corpus and presents correct expressions along with incorrect expressions using a grammatical error detection system so that the learner does not need to be aware of how to search for examples. Intrinsic and extrinsic evaluations indicate that our method improves the accuracy of example sentence retrieval and the quality of a learner's writing.

1 Introduction

Grammatical error detection for learners of English as a second language (ESL) is widely studied. However, there are few studies on grammatical error detection for learners of Japanese as a second language (JSL). Most studies on grammatical error detection in Japanese focus on a learner's particular error types, mainly with particles (Suzuki and Toutanova, 2006; Imamura et al., 2012). Among others, there are studies using phrase-based statistical machine translation (PBSMT), which does not limit the types of grammatical errors made by a learner (Mizumoto et al., 2011). However, PBSMT-based grammatical error detection cannot consider long-distance relationships because it relies on either character or word n-grams.

A standard method that supports the effort of learning a second language is the use of examples. Example retrieval systems such as Rakhilina et al. (2016) and Kilgarriff et al. (2004) in particular check for the appropriate use of words based on the context in which they are written. However, in such a system, if the query word is incorrect, finding appropriate examples is impossible using ordinary search engines, such as Google. Even if learners have access to an incorrect example retrieval system, such as Kamata and Yamauchi (1999) and Nishina et al. (2014), they do not know how to search for the examples because they do not know whether their query includes errors. Moreover, they are often unable to rewrite a composition in the absence of correct versions of the incorrect examples. These systems are primarily developed for use by Japanese teachers. As such, they are not as helpful for learners who do not have a strong background in Japanese.

Considering this, our study develops an example sentence retrieval system[1] with grammatical error detection using the large-scale Lang-8[2] dataset for JSL by focusing on the usability of automatic incorrect example retrieval. The main contributions of this work are as follows:

- This is the first study that tackles grammatical error detection in Japanese using a neural network. It shows the state-of-the-art F score on the Lang-8 dataset and establishes a new baseline.
- To the best of our knowledge, our system is the first incorrect example sentence retrieval system using neural grammatical error detection. This function allows a user to recognize which part of the query is wrong.
- Our system seamlessly shows the incorrect sentences, and the corresponding sentences corrected by a native speaker. Thus, learners

[1] http://cl.sd.tmu.ac.jp/sakura/v3

[2] Multi-lingual language learning and language exchange social networking service. http://lang-8.com/

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 296–305
Florence, Italy, August 2, 2019.

Name	Correct Sent.	Incorrect Sent.	Revised Sent.	Error Detection
Learners' Error Corpora of Japanese Searching Platform	✓	✓	✓	×
Tagged KY corpus	✓	✓	×	×
Proposed system	✓	✓	✓	✓

Table 1: Features of example retrieval systems for Japanese language learners. "Correct Sent." indicates whether the system can display the correct sentences; "Incorrect Sent." indicates whether the system can display the incorrect sentences; "Revised Sent." indicates whether the system can display the revised sentence corresponding to the incorrect sentence; and "Error Detection" denotes whether the system has a grammatical error detection system.

can rectify their mistakes while writing the composition.

- Our intrinsic evaluation shows that our system is good at correcting lexical choice and misformation errors in a learner's writing. Our extrinsic evaluation also shows that our example sentence retrieval system improves the quality of a learner's writing.

2 Related Works

2.1 Grammatical Error Detection

In the grammatical error detection task in English, neural methods such as Bi-LSTM in particular have been actively used (Rei et al., 2016; Rei and Yannakoudakis, 2016; Kaneko et al., 2017; Kasewa et al., 2018). Most studies on grammatical error detection/correction in Japanese limit the target learner's error types, mainly to particles (Imaeda et al., 2003; Suzuki and Toutanova, 2006; Imamura et al., 2012; Oyama et al., 2013). Among others, there are studies in Japanese grammatical error correction using statistical machine translation which do not limit the type of errors from the learner (Mizumoto et al., 2011). On the other hand, in Japanese, there are few studies on grammatical error detection and correction using neural networks.

In this study, we constructed an error detection system using a neural network without limiting the target error type. Although phrase-based statistical machine translation cannot consider long-distance relationships because it is n-gram based, neural networks using Bi-LSTM can consider long-distance relationships because they can maintain input history. Recently, neural network-based approaches outperformed PBSMT-based methods in grammatical error correction (Junczys-Dowmunt et al., 2018; Chollampatt and Ng, 2018); they are expected to be effective in grammatical error detection as well.

2.2 Example Retrieval System for Japanese as a Second Language

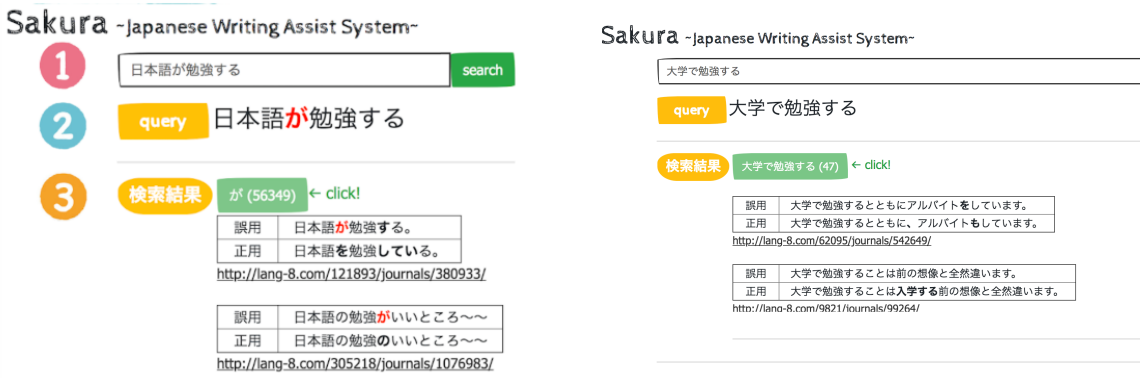
Various Japanese example retrieval systems were proposed in recent times. However, in practice, learners find them difficult to use. We explain herein the reasons why these systems are not effective when used by JSL learners.

Table 1 lists the features of each system. Our proposed system, Sakura, employs a large-scale Japanese JSL corpus for correct and incorrect example sentences along with revisions for the incorrect example.

First, the "Learner's Error Corpora of Japanese Searching Platform"[3] was constructed by the Corpus-based Linguistics and Language Education at Tokyo University of Foreign Studies. This system displays sentences in the keyword in context (KWIC) format based on the learner's information, such as native language, age, and gender. Japanese language teachers can identify the features of the learner's mistakes using this system. However, this system is primarily intended for educators rather than learners. As such, learners might find it confusing to use. In addition, this system has few examples. Also, the users may not know whether their query includes errors because it does not perform grammatical error detection. Therefore, they do not know how to search for the examples.

Second, the "KY corpus" is a transcribed speech corpus for JSL learners. "Tagged KY corpus" (Kamata and Yamauchi, 1999) supersedes the "KY corpus" with a search engine using POS. It displays correct and incorrect examples for text written by learners. However, it has the drawback that often, no results are provided, even for high-frequency words, because the number of incorrect examples is small; therefore, it is difficult for language learners to use the limited set of examples as a reference.

[3] http://ngc2068.tufs.ac.jp/corpus_ja/



(a) incorrect

(b) correct

Figure 1: User interface of our system.

sentence	いま	、	ぼく	は	がっ	く	が	とても	いそがし	です	よ	。
label	c	c	c	c	c	**i**	c	c	**i**	c	c	c

Table 2: Example of incorrect and correct labels. The *c* indicates that the target word is correct. The *i* indicates that the target word is incorrect. The meaning of this sentence is "I am very busy at school now."

2.3 Example Retrieval System for English as a Second Language

Web-based search engines are the most common search systems that can be used to search for example sentences. However, these search engines are not intended to retrieve examples for language learners; therefore, the search engines neither show example sentences nor the correct version of a given incorrect sentence to aid learners.

Language learners can use several example retrieval systems. All of them provide special features for writing assistance, but none of them offers grammatical error detection and incorrect examples to support learners.

FLOW (Chen et al., 2012) is a system that displays some candidates for English words when ESL learners write a sentence in their native language using candidate paraphrases with bilingual pivoting. By contrast, our system suggests incorrect examples and their counterparts based on corrections from the learner corpus.

Another system, called StringNet (Wible and Tsao, 2010), displays the patterns in which a query is used, along with their frequency. The noun and the preposition are substituted by their parts of speech, in place of the words themselves, to eliminate data sparseness.

The ESCORT (Matsubara et al., 2008) system shows example sentences to learners based on the grammatical relations of queries. The syntactic structures of the English sentences are stored in the database of a raw corpus. ESCORT analyzes the dependency relations of the input queries and only displays appropriate examples that match the relations. Our system displays the examples in descending order of the cosine similarity of the input vector and vectors of the examples to avoid data sparseness.

Furthermore, ESL learners can check examples while writing an English sentence using WriteAhead (Yen et al., 2015). This system shows pattern suggestions based on collocation and syntax. For example, when the user writes "We discussed," the system displays the patterns for the use of the word "discussed."

Sketch Engine (Kilgarriff et al., 2004) displays the grammar constructs associated with words along with the thesaurus information. As previously mentioned, our system presents incorrect examples using a learner corpus apart from the correct examples extracted from a raw corpus.

3 Incorrect Example Retrieval System using Grammatical Error Detection for JSL

This section describes our incorrect example retrieval system with grammatical error detection. It combines grammatical error detection and example sentence retrieval. We assume that language learners put queries that may contain errors so that we will perform grammatical error detection on the users' input. If errors are detected, it will be

passed to the incorrect example sentence search; otherwise, it will be processed by the correct example sentence search.

This section is organized as follows. Section 3.1 shows the typical use case of our system. The user interface illustrated in Section 3.2 allows learners to search for incorrect examples. The grammatical error detection algorithm is explained in Section 3.3. Our example sentence retrieval algorithm is explained in Section 3.4.

3.1 Use Case

One of the obstacle in learning Japanese as a second language is to learn the use of particles. Particles in Japanese indicate grammatical relations between verbs and nouns. For example, the sentence, "日本語を勉強する。", which means "I study Japanese." includes an accusative case marker "を", which introduces the direct object of the verb. However, in this case, Japanese learners often make mistakes, such as "日本語が勉強する。", which means "Japanese language studies." Thus, the appropriate use of particles is not obvious for non-native speakers of Japanese. Particle errors and word choice are the most common Japanese grammatical errors (Oyama et al., 2013), both of which require a sufficient number of correct and incorrect examples to understand the usage in context. A word n-gram search provides only a few or no examples for a phrase because Japanese is a relatively free word order language, in which a syntactically dependent word may appear in a distant position.

Ideally, Our system can deal with these particle errors. Figure 1 (a) illustrates an example of the search result obtained using our system. Suppose a learner wants to view examples for the usage of "日本語が勉強する (*nihongo **ga** benkyousuru*, which include an incorrect usage "が"(*ga*))". As can be seen in No.2 of Figure 1, our system indicates the query with "が" written in red. The learner can recognize that "が" is wrong. As can be seen in No.3 of Figure 1, our system displays correct examples using "日本語を勉強する。(*nihongo **wo** benkyousuru*, which is the correct euphonic form of "I study Japanese")". The learner can then identify that "が" is the incorrect word, and "を" is the correct word.

If the query returns that learner input is correct, our system shows the examples that match the query. For example, Figure 1 (b) displays the examples using "大学で勉強する (*daigaku de benkyou suru*, meaning "I study at university.") which is the correct sentence.

3.2 User Interface

Figure 1 shows the user interface of our system. There are two types of user interfaces. Figure 1 (a) and Figure 1 (b) show the example search interfaces used when searching for incorrect and correct examples, respectively. The components of the user interface are explained below.

1. Query Input the words to be searched for. The input query is assumed to be a sentence or several words (a sequence of words).

2. Grammatical error detection The system detects errors. If errors are detected, the part with errors is displayed in red.

3. Retrieval result The retrieval results that match the query are displayed. The incorrect sentences written by learners are shown in the upper part, paired with the correct examples revised by native speakers. The revised part is represented in bold.

3.3 Grammatical Error Detection

In this study, grammatical error detection is treated as a sequence labeling task and each word in the input sentence is assigned an incorrect or correct label. Table 2 shows the example of labels. We labeled detection tags using dynamic programming from incorrect sentences and correct sentences.

We used the character- and word-level Bi-LSTM models for grammatical error detection, proposed by Rei et al. (2016). As with Rei et al. (2016), we construct a concatenation-based character-level Bi-LSTM and word-level Bi-LSTM for error detection. Our code is available on GitHub[4]. Figure 2 is the construction of our model. The system receives words $[w_1...w_T]$ as input and predicts labels for each word. A word w_t is converted to word vector e_t^{w} and character vector e_t^{c} using word-leve Bi-LSTM and character-level Bi-LSTM, respectively. The character vector is created by combining the hidden states of the beginning and the end of character-level Bi-LSTM, which takes one character as input.

[4]https://github.com/kanekomasahiro/japanese_error_detection

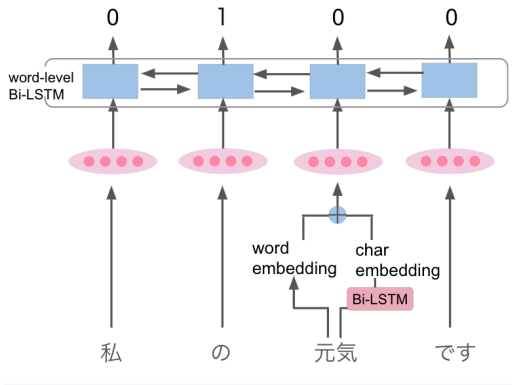


Figure 2: Architecture of our grammatical error detection.

The t-th input vector $\tilde{x}_t$ is created by combining e_t^{w} and e_t^{c}. The input vector calculates the hidden states h_t as follows using a character- and word-level Bi-LSTM (Hochreiter and Schmidhuber, 1997):

$$\overrightarrow{h}_t = LSTM(\tilde{x}_t, \overrightarrow{h}_{t-1}) \quad (1)$$

$$\overleftarrow{h}_t = LSTM(\tilde{x}_t, \overleftarrow{h}_{t+1}) \quad (2)$$

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \quad (3)$$

$\overrightarrow{h}_t$ is forward LSTM, $\overleftarrow{h}_t$ is backward LSTM, and h_t is a combination of hidden states in both directions. We calculate an additional hidden layer d_t to mitigate the dimensionality difference between Bi-LSTM and the output layer using the full connected layer:

$$d_t = \tanh(W_{\mathrm{d}} h_t) \quad (4)$$

W_{d} is a weight matrix. We make predictions using the output layer and the softmax function:

$$P(y_t|w_1 ... w_T) = \mathrm{softmax}(W_{\mathrm{o}} d_t) \quad (5)$$

W_{o} is an output weight matrix and y_t is a prediction label.

3.4 Example Sentence Retrieval Algorithm

The input queries can be either sentences or words. Once the user enters a query, examples of incorrect sentences written by language learners and their corresponding correct sentences are retrieved from the learner corpus and displayed in pairs. The search strategy is described below:

1. The error detection model processes an input query.
2. If an error is detected, the error part of the query is searched from examples of incorrect sentences. The incorrect examples are displayed along with their correct examples in descending order of the cosine similarity[5] of the input vector and vectors of the examples of incorrect sentences.
3. If no error is detected, the entire query is searched among the examples of correct sentences. The system displays the incorrect examples along with their correct versions in descending order of the cosine similarity of the input vector and vectors of the examples of correct sentences.

4 Experiments

4.1 Dataset

In this study, we use the Lang-8 Learner Corpora created by Mizumoto et al. (2011). The developers of the dataset used it for Japanese grammatical error correction, whereas we used it as an example retrieval database for JSL.

Each learner's sentence has at least one revised sentence. A learner's sentence is combined with a revised sentence to make a sentence pair. If a learner's sentence has more than one revised sentence, each of the revised sentences is paired with the learner's sentence as separate sentence pairs. Sentences with a length of more than 100 words or with a Levenshtein distance of more than 7 are eliminated to remove the noise in the corpus.

We extracted 1.4 million pairs of learner sentences written by Japanese language learners and revised sentences corrected by Japanese native speakers. The total number of included Japanese essays was 185,991.

The learner sentences and the revised sentences were tokenized by the morphological analyzer, MeCab (ver. 0.996)[6] with UniDic (ver. 2.2.0). We used `gensim`[7] to create the sentence vectors.

4.2 Grammatical Error Detection

For the experiments with error detection, we use the dataset described in Section 4.1. We split the corpus into 720,000 sentences for training data, 1,000 sentences for development data, and 1,000

[5] We use word2vec (Mikolov et al., 2013) to obtain the word vectors. The average of the word vectors is taken as a sentence vector.

[6] https://github.com/taku910/mecab

[7] https://github.com/RaRe-Technologies/gensim

model	precision	recall	F-value$_{0.5}$
SMT system	0.599	0.121	0.202
proposed system	**0.615**	**0.304**	**0.407**

Table 3: Accuracy of detection of writing errors made by Japanese learner.

error type	TP	FN	FP
all	294	263	106
particle choice	**75**	60	
alternating form	16	38	
lexical choice	22	77	
omission	**33**	18	
misformation	**53**	14	
redundant	**40**	27	
pronunciation	**55**	25	
others	0	4	

Table 4: Number of true positives, false negatives, and false positives. "TP", "FN", and "FP" indicate true positive, false negative, and false positive, respectively.

sentences for test data, respectively. We used automatically converted error tags as the gold label for grammatical error detection.

Setting For hyper parameter settings, the dimension of the word embedding and the word-level LSTM are 300, and the dimension of the character embedding and the character-level LSTM are 100. The Bi-LSTM models are optimized using Adadelta with a learning rate of 1.0 and a batch size of 64 sentences. These word and character embeddings are updated during training.

We reimplemented the word-wise phrase-based statistical machine translation system of Mizumoto et al. (2011) as a baseline system. We used minimum error rate training (MERT) (Och, 2003) for the model.

Result The results are shown in Table 3. It can be seen that all of the precision, recall, and F-values are better than the baseline. As Nagata and Nakatani (2010) suggested, a high precision error detection system can be used to help learners write essays. We will verify this hypothesis in the next subsection.

Table 4 lists the number of true positives and false negatives by error type. Particle choice, pronunciation, and misformation are easy to detect. Lexical choice and alternating form are hard to detect. The number of false negatives for particle choice is large because it forms the majority of all the errors.

Table 5 shows the example sentences detected as true positives and false negatives by our method. Because of the neural network, our method can detect a long-distance error such as the column of "true positive" in Table 5. "お願い" (meaning, "Please.") is at the beginning of the sentence, because of which this sentence is not considered to be of future tense; instead, it is considered as expressing desire. Therefore, it can be seen that it is appropriate to use "～たい" to mark desire explicitly. LSTM can deal with this kind of long-distance dependency; hence, our method can detect such errors. On the other hand, the column of "false negative" on Table 5 shows that the learner has incorrectly input "家康" as "家安". "家康" is the name of a famous historical personage, and its misspelled variant, "家安", is not in the data. The column of "false positive" on Table 5 is an example of misdetection. "名刺" is a noun and the corpus has only two instances of "名刺", which co-occur with a different particle "に" (dative case marker). Such errors cannot be detected owing to lack of data.

4.3 Incorrect Example Retrieval System

Intrinsic Evaluation We randomly extracted 55 incorrect phrases and 55 correct phrases from the learner's sentences in the Lang-8 dataset, which are not included in the corpus of the retrieval system. We classified each incorrect example into seven types: alternating form (A), lexical choice (L), omission (O), misformation (M), redundant (R), pronunciation (P), and others (X). Table 6 lists the examples of the test phrases.

Table 7 shows the frequency of each error type and the relevance of each system per error type. An example is judged relevant if it matches the auto-tagged results annotated to the data; otherwise, it is judged irrelevant. Because the user needs to select whether to search correct examples or incorrect examples in previous work, both the baseline correct example retrieval system (BC) and the baseline incorrect example retrieval system (BI) are used as the baseline systems. We searched for these phrases in each system (BC, BI, and ours) and counted the number of hits for each system that led to the top-1 correct expressions to measure relevance. The proposed system searches either the correct or incorrect sentences including the target phrase depending on whether the query contains errors while it searches for the phrase cor-

true positive	incorrect	おねがい 、 しあわせ に なる ！
	correct	おねがい 、 しあわせ に なり たい ！
	meaning	Please, I hope to be happy!
false negative	incorrect	定刻 に なる と 、 徳川 家安 が 出 て き ます 。
	correct	定刻 に なる と 、 徳川 家康 が 出 て き ます 。
	meaning	When the time comes, Tokugawa Ieyasu will be coming.
false positive	correct	これ 、 私 の 名刺 で ござい ます 。
	meaning	This is my business card.

Table 5: Examples of true positive, false negative, and false positive.

incorrect phrase	pronunciation	correct phrase	pronunciation	BC	BI	**Ours**	type
おねさん	onesan	おねえさん (sister)	oneesan	×	×	✓	O
ニュージランド	nyu-jirando	ニュージーランド (New Zealand)	nyu-ji-rando	×	✓	✓	O
みんなさん	min'nasan	みなさん (eveybody)	minasan	×	✓	✓	R
大体に	daitaini	大体 (roughly)	daitai	×	×	✓	R
疑問をして	gimonwoshite	疑問に思って (in doubt)	gimon'niomotte	×	×	✓	M
驚い	odoroi	驚き (surprise)	odoroki	×	✓	×	M
がもらえる	gamoraeru	しかもらえない (only get this)	shikamoraenai	×	×	×	A
稼ぐ	kasegu	稼いだ (earned)	kaseida	×	✓	×	A
ちさい	chisai	少ない (few)	sukunai	×	✓	×	L
助けられる	tasukerareru	できる (can)	dekiru	×	×	✓	L
しましだ	shimashida	いました (there was)	imashita	×	×	×	P
死んちゃう	shincha	死んじゃう (will die)	shinjau	×	✓	✓	P
ハウス	hausu	家 (house)	ie	×	✓	✓	X

Table 6: Examples of test results. The column "Incorrect phrase" contains the phrases written by the learner. These are extracted from the Lang-8 test set. The column "Ours" shows whether our system was able to find the correct answer for that phrase.The column "type" shows the error type of each phrase.

error type	frequency	relevance		
		BC	BI	Ours
incorrect all	55	0.00	**0.45**	0.44
alternating form	19		**0.37**	0.32
lexical choice	16		**0.38**	0.19
omission	8		**0.75**	**0.75**
misformation	6		0.40	**0.67**
redundant	3		0.67	**1.00**
pronunciation	2		**0.50**	**0.50**
others	1		**1.00**	**1.00**
correct	55	**0.90**	0.15	0.85
average	110	0.45	0.30	**0.65**

Table 7: Frequency and relevance of each system (intrinsic evaluation).

responding each system.

The BC system has the highest relevance for correct phrases, but has no matches for incorrect phrases; therefore, the relevance becomes 0.00 in the incorrect example retrieval task. BI, on the other hand, finds almost no examples when searching for correct phrases, while high relevance is obtained with incorrect phrases. In our proposed system, although the overall relevance of incorrect phrases is little lower than that of BI, the user has to switch between the incorrect retrieval and the correct retrieval in the baseline systems. The proposed system determines whether the query is correct by using error detection. This system gets the highest overall relevance, including for both the incorrect phrase and correct phrase retrieval tasks.

In contrast to the baseline system, the proposed system can detect misformation well. Because the erroneous expression is explicit in this error type, the accuracy of error detection is high and it presents relevant sentences at the top. In addition, obvious errors such as omission and redundancy are easily detected, so it receives a high relevance rate.

On the other hand, searching for a lexical choice is difficult. If the sentences written by the learner are syntactically correct but semantically incorrect, the system cannot detect errors. Additionally, because the recall of error detection is not sufficient, it sometimes misses an incorrect input query and searches through correct examples.

Extrinsic Evaluation In the extrinsic evaluation, we compared the writing scores of compo-

No.	Prompt
1	Introduce the city you live in.
2	Which do you like better, summer or winter?
3	Which aspects of Japanese do you find difficult?
4	What is the difference between televised and printed news?
5	What would you like to experience overseas?
6	If you have free time, what would you like to do?
7	Introduce the charm of your country.
8	Is it a good thing to tell a lie?
9	What are you doing for your health?
10	What was the most enjoyable thing in university life?

Table 8: Prompts for extrinsic evaluation.

Learner	BC w/o ED	BC+BI w/o ED	BC+BI w/ ED
A	14	20	**21**
B	26	27	**29**
C	15	**16**	**16**
D	**28**	25	26
E	22	**25**	**25**
F	20	23	**28**
ave.	20.8	22.7	**24.2**

Table 9: Result of extrinsic evaluation.

sitions using three systems. All systems used the data constructed in Section 4.1.

- BC w/o ED: Perform no error detection and search correct examples only.
- BC+BI w/o ED: Perform no error detection and search correct examples and incorrect examples according to the user's choice.
- BC+BI w/ ED: Perform error detection and search for correct and incorrect examples automatically.

We compare the writing score of the composition using the BC system w/o ED against the that of the BC+BI system w/o ED to confirm the usability of incorrect examples. We compare the writing score of the composition using BC+BI systems with and without ED to check the practicality of the error detection module.

We recruited six Japanese non-native speakers majoring in computer science in a graduate school in Japan to complete 10 Japanese composition exercises. The prompts of the 10 Japanese composition exercises are shown in Table 8. Chinese was the native language of all participants. Five of the participants had passed the N1 (advanced) Japanese-Language Proficiency Test, while the other had passed the N2 (intermediate) level. We divided the prompts into five prompts each and asked each learner to write either of the halve using the BC system w/o ED and the other half using BC+BI system w/o ED. After that, they were asked to use the BC+BI system w/ ED to revise the composition. The number of sentences in each exercise was three to ensure a fair comparison. The composition exercise was given a score by deducting points, and each participant was assigned 30 points at the beginning. One point was deducted per error. The total score of each system was taken over five exercises.

error type	frequency	#	relevance
incorrect all	44	9	0.20
alternating form	6	1	0.17
lexical choice	14	4	**0.29**
omission	7	1	0.14
misformation	6	2	**0.33**
redundant	9	1	0.11
pronounciation	2	0	0.00
other	0	0	0.00

Table 10: Frequency and relevance of our system for an actual learner's composition (extrinsic evaluation).

The results of the extrinsic evaluation are shown in Table 9. We confirmed that the highest score was achieved using the proposed system, and 5 out of the 6 people achieved the highest score using the proposed system.

Table 10 shows the ratio of errors that could be corrected when the compositions were first written using the BC+BI system w/o ED and then revised using the BC+BI system w/ ED. We manually checked all errors and classified them as relevant or irrelevant. As with intrinsic evaluation, misformation was corrected at the highest rate. Unlike intrinsic evaluation, lexical choice was corrected well, but it can be seen from the breakdown that function words can also be corrected at a high rate. The relevance of suggestion of lexical choice for content words was 0.17 whereas that for function words was 0.38. It was not clear from the intrinsic evaluation because function words such as particles are not the targets of evaluation, but it is understood that a neural grammatical error detection method can cope with lexical choice errors for function words such as particles frequently found in writings by learners.

5 Conclusion

We constructed a large-scale incorrect example retrieval system with grammatical error detection for JSL learners. Our proposed system switches between incorrect example sentence retrieval and correct example sentence retrieval automatically by using grammatical error detection and then displays incorrect examples along with the revised sentences and example sentences. The results of our experiment showed that our system was useful for JSL learners in writing Japanese compositions. Each example includes incorrect sentences; hence, language teachers can identify the difficulty faced by learners and use this information for language education.

Although this system was constructed for JSL learners, it can easily be customized for other languages. We plan to extend our system to support ESL learners (Tajiri et al., 2012).

Acknowledgements

We would like to thank the Lang-8 web organizer for providing the text data for our system. This work was partially supported by JSPS Grant-in-Aid for Scientific Research (C) Grant Number JP19K12099.

References

Mei-Hua Chen, Shih-Ting Huang, Hung-Ting Hsieh, Ting-Hui Kao, and Jason S. Chang. 2012. FLOW: A first-language-oriented writing assistant system. In *Proceedings of the ACL 2012 System Demonstrations*, pages 157–162.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of AAAI*, pages 5755–5762.

Sepp Hochreiter and Jŭrgen Schmidhuber. 1997. Long short-term memory. In *Neural Computation, 9*.

Koji Imaeda, Atsuo Kawai, Yuji Ishikawa, Ryo Nagata, and Fumito Masui. 2003. Error detection and correction of case particles in Japanese learner's composition. In *Proceedings of the Information Processing Society of Japan SIG*, pages 39–46.

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and Hitoshi Nishikawa. 2012. Grammar error correction using pseudo-error sentences and domain adaptation. In *Proceedings of ACL*, pages 388–392.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of ACL*, pages 595–606.

Osamu Kamata and Hiroyuki Yamauchi. 1999. KY corpus version 1.1. Report, Vocaburary Acquisition Study Group.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error- and grammaticality-specific word embeddings. In *Proceedings of IJCNLP*, pages 40–48.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of EMNLP*, pages 4977–4983.

Adam Kilgarriff, Pavel Rychly, Pavel Smrž, and David Tugwell. 2004. The sketch engine. In *Proceedings of EURALEX*, pages 105–116.

Shigeki Matsubara, Yoshihide Kato, and Seiji Egawa. 2008. ESCORT: example sentence retrieval system as support tool for English writing. *Journal of Information Processing and Management*, 51(4):251–259.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

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.

Ryo Nagata and Kazuhide Nakatani. 2010. Evaluating performance of grammatical error detection to maximize learning effect. In *Proceedings of COLING*, pages 894–900.

Kikuko Nishina, Bor Hodošček, Yutaka Yagi, and Takeshi Abekawa. 2014. Construction of a learner corpus for Japanese language learners: Natane and Nutmeg. *Acta Linguistica Asiatica*, 4(2):37–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Hiromi Oyama, Mamoru Komachi, and Yuji Mastumoto. 2013. Towards automatic error type classification of Japanese language learners' writing. In *Proceedings of PACLIC*, pages 163–172.

Ekaterina Rakhilina, Anastasia Vyrenkova, and Elmira Mustakimova. 2016. Building a learner corpus for Russian. In *Proceedings of the Joint Workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, pages 66–75.

Marek Rei, Gamal K.O. Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. In *Proceedings of COLING*, pages 309–318.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of ACL*, pages 1181–1191.

Hisami Suzuki and Kristina Toutanova. 2006. Learning to predict case markers in Japanese. In *Proceedings of ACL*, pages 1049–1056.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of ACL*, pages 198–202.

David Wible and Nai-Lung Tsao. 2010. StringNet as a computational resource for discovering and investigating linguistic constructions. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, pages 25–31.

Tzu-Hsi Yen, Jian-Cheng Wu, Jim Chang, Joanne Boisson, and Jason Chang. 2015. WriteAhead: Mining grammar patterns in corpora for assisted writing. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 139–144.