Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners

Tomoya Mizumoto
NAIST, Japan
tomoya-m@is.nasit.jp

Mamoru Komachi
NAIST, Japan
komachi@is.naist.jp

Masaaki Nagata
NTT, Japan
nagata.masaaki@lab.ntt.co.jp

Yuji Matsumoto
NAIST, Japan
matsu@is.naist.jp

Abstract

We present an attempt to extract a large-scale Japanese learners' corpus from the revision log of a language learning SNS. This corpus is easy to obtain in large-scale, covers a wide variety of topics and styles, and can be a great source of knowledge for both language learners and instructors. We also demonstrate that the extracted learners' corpus of Japanese as a second language can be used as training data for learners' error correction using an SMT approach. We evaluate different granularities of tokenization to alleviate the problem of word segmentation errors caused by erroneous input from language learners. Experimental results show that the character-wise model outperforms the word-wise model.

1 Introduction

The number of Japanese language learners around the world has increased more than 30-fold in the past three decades. The Japan Foundation reports that more than 3.65 million people in 133 countries and regions are studying Japanese in 2009 [1]. However, there are only 50,000 Japanese language teachers overseas, and thus it is in high demand to find good instructors for writers of Japanese as a Second Language (JSL).

Recently, natural language processing research has begun to pay attention to second language learning (Rozovskaya and Roth, 2011; Park and Levy, 2011; Liu et al., 2011; Oyama and Matsumoto, 2010; Xue and Hwa, 2010). However, most previous research for second language learning deals with restricted types of learners' errors. For example, research for JSL learners' errors mainly focus on Japanese case particles (Oyama and Matsumoto, 2010; Imaeda et al., 2003; Nampo et al., 2007; Suzuki and Toutanova, 2006), however they focus only on case particles whereas we attempt to correct all types of errors.

However, real JSL learners' writing contains not only errors of Japanese case particles but also various other errors including spelling and collocation errors. For instance, a Japanese language learner who speaks Chinese may write:

何で日本語はこんなに難しいなの？
(Why does Japanese are so difficult?)

which has a grammatical error of inserting 'な' due to literal translation from Chinese. Park and Levy (2011) proposed an EM-based unsupervised approach to perform whole sentence grammar correction, but the types of errors must be predetermined to learn the parameters for their noisy channel model. It requires expert knowledge of L2 teaching, which is often hard to obtain.

One promising approach for correcting unrestricted errors of JSL learners is Brockett et al. (2006)'s automated error correction method using statistical machine translation (SMT). The advantage of their method is that it does not require expert knowledge. Instead, it learns a correction model from sentence-aligned corrected learners' corpora. However, it is not easy to acquire large-scale learners' corpora. In fact, Brockett et al. (2006) used regular expressions to automatically create erroneous corpora from native corpora.

To solve the knowledge acquisition bottleneck, we propose a method of mining revision logs to create a large-scale learners' corpus. The corpus is compiled from error revision logs from a language learning social network service (SNS), which covers a wide variety of topics and styles. The main advantage of using revision logs is three-fold: (1) it benefits from the wisdom of crowds, (2) it can be obtained in large-scale, and (3) it is a great source

[1] http://www.jpf.go.jp/e/japanese/survey/result/index.html

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 147–155,
Chiang Mai, Thailand, November 8 – 13, 2011.

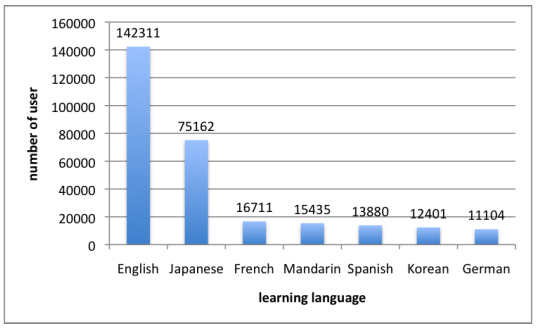

Figure 1: Number of users for each learning language in Lang-8

of knowledge not only for learners but also for language teachers. In this paper, we show that the method using SMT techniques with a large-scale learners' corpus can correct JSL learners' error with reasonable accuracy.

The rest of this paper is organized as follows. Section 2 describes the JSL corpus created from revision logs of a language learning SNS. Section 3 explains an SMT-based approach to JSL error correction. In section 4, we report the experimental results of SMT-based JSL error correction using large-scale real corpus. In section 5, we conduct error analysis and discuss issues of the proposed method. Section 6 concludes this work and presents future direction.

2 A Large Scale Japanese Language Learners' Corpus from Revision Logs

Recent growth of the web has opened the possibility of using the internet to break the barriers of space and time. Specifically, social network service (SNS) has begun to receive a lot of attention recently. There are a number of SNS sites that help language learners across the world, including smart.fm, Livemocha and Lang-8, to name a few. We will look briefly at each SNS below.

First, smart.fm [2] (formerly iKnow!) is a SNS-based language learning service that helps learners practice language learning. Smart.fm provides a tailored curriculum for each user to memorize learning content through simple exercises.

Second, Livemocha [3] is also a language learning SNS that offers course of grammar instructions, reading comprehension exercises and practice for both writing and speaking. Users can submit a free composition on a subject and receive feedbacks from other users of the native language. However, they are not able to write about arbitrary topics.

Third, Lang-8 is a "Multi-lingual language learning and language exchange Social Networking Service" [4] , which has over 200,000 registered members at the moment. As soon as the learners write a passage, mostly a part of a diary, in a language they are learning, native speakers of the language correct it for them. The learners in turn are encouraged to correct other members' composition errors according to their first language (L1). Hence, the SNS is called "language exchange". It supports 77 languages, facilitating multilingual communication.

2.1 Japanese Language Learners' Corpora

One of the most famous learners' corpus is Teramura Error Data [5]. The corpus was mainly collected in 1986 from Japanse compositions written by foreign students, mostly from Asian countries. The corpus consists of several styles including free composition, cloze (gap filling) test, and pattern composition. Unlike this data, JSL learners in Lang-8 encompass the whole world. Also, Lang-8 offers a wide variety of free compositions of the learner's choice, and the size of the data is orders of magnitude (448MB without all the tags) larger than Teramura's data (420KB, 4,601 sentences written by 339 students). Also, although Teramura Error Data is annotated with error types, the correct words or strings are not often provided, which makes it difficult to use it for automatic correction of learners' errors.

Ohso [6] created a database of Japanese compositions by JSL learners. It is annotated with error types with correct forms to allow error analysis. However, similar to Teramura Error Data, the corpus does not cover many topics because it was collected at only four institutions. In addition, it is limited in size (756 files, average file size is 2KB).

The corpus most related to ours is the JSL learners parallel database of Japanese writings and their translation of learners' L1 [7] created by National Institute for Japanese Language and Linguistics. It collects 1,500 JSL learners' writings and their

[2] http://smart.fm/
[3] http://www.livemocha.com/
[4] http://lang-8.com/
[5] http://www.lang.nagoya-u.ac.jp/tonoike/teramura.html
[6] https://kaken.nii.ac.jp/ja/p/08558020/1998/6/en
[7] http://jpforlife.jp/taiyakudb.html

self-translations. There are around 250 writings corrected with error types by several Japanese language teachers. The advantage of this corpus is that some of the texts are annotated by professional language teachers and can be used as a source of error correction. However, again, the size of this corpus is limited since it is hard to obtain annotations from language teachers. Our approach differs from them in that we employ the wisdom of crowds of native speakers, not necessarily language teachers, to compile a large-scale learners' corpus.

2.2 Features of Lang-8 Data

We created a large-scale language learners' corpus from error revision log of Lang-8. Figure 1 shows that approximately 75,000 users are learning Japanese [8]. Table 1 shows the top seven languages in the corpus. There are 925,588 sentences of JSL learners [9]. Out of 925,588 sentences, 763,971 (93.4%) sentences are corrected by human annotators. A sentence written by JSL learners might have two or more revision sentences in Lang-8 by different voluntary reviewers [10]. Therefore, the total number of corrected sentences amounts to 1,288,934. In other words, one sentence gets corrected approximately 1.69 times on average.

There are several distinguishing features of the data obtained from Lang-8. First, since Lang-8 is a language learning SNS, we can obtain pairs of learner's sentence and corrected sentence. Using this data, it is possible to collect the learner's errors. We will describe how to build a learners' corpus from revision logs later in this section.

Second, Lang-8 data may have more than one correction for the same sentence. We could exploit this feature to acquire paraphrases in a similar way to (Barzilay and McKeown, 2001). Table 2 shows an example of multiple correction. Two annotators correct the same learner's sentence. In this example, one can infer that "なりの表現で (in one's own expressions)" and "なりに (in one's own way)" are paraphrases of each other.

Third, we could obtain multi-lingual parallel sentences. Figure 2 shows examples of parallel

Figure 2: Parallel sentence in Lang-8

sentences in Lang-8. In this example, the JSL learner writes two Japanese sentences and their translation for each sentence to tell what he or she wants to say. Although the sentences written in the learning language may contain errors and mistakes, we can align the English translation to the corrected Japanese sentence. The parallel corpus created from the revision log of SNS would be a remarkable source of colloquial expressions ideal for translating consumer generated media such as blogs and SNS.

Fourth, annotators of Lang-8 sometimes add inline comments to the corrected sentences. It is often written in parentheses to indicate that the string is a comment, but not always. Depending on the first language of the language learner, annotators put comments in either the learning language or the learner's L1. This can be a great source of extracting useful information for language learning, since the comment itself explains pitfalls that the language learners often come across.

2.3 Extracting corrected sentence from HTML

All the error revisions are made through a web-based editing interface that allows annotators to delete, insert or change any character sequence of the learner's text by any sequence. Table 3 illustrates an example of the HTML generated from Lang-8's revision editor. The tag `<span class="sline">` shows that the characters within the tags should be removed. The color tags `<span class="red">` and `<span class="f_blue">` are used somewhat arbitrarily by annotators. In general, they indicate correct strings. In the example, the annotator used delete line and red color to point out and correct the first error, and blue color to indicate inserted characters.

From this observation, we apply simple

[8] We counted learning language in user profile. Some learners register two or more learning languages.

[9] We counted learning language written for each journal because learners may write in different languages.

[10] The correction of a new review might be affected by the previous corrections by others.

Table 1: Number of sentences for each language in Lang-8

Language	English	Japanese	Mandarin	Korean	Spanish	French	German
Number of sentences	1,069,549	925,588	136,203	93,955	51,829	58,918	37,886

Table 2: An illustraive example of multiple correction

Sentence written by a JSL learner	三人はそれぞれ自分 の方式で 感情を表 れ ます。
Sentence corrected by an annotator1	三人はそれぞれ自分 なりの表現で 感情を表 し ます。 (Each of three expresses their feelings in their own expressions.)
Sentence corrected by an annotator2	三人はそれぞれ自分 なりに 感情を表 し ます。 (Each of three expresses their feelings in their own way.)

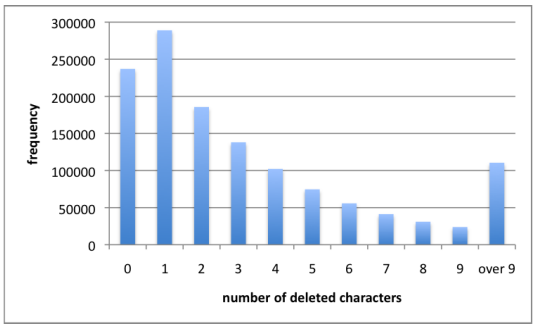

Figure 3: Summary of number of deletion

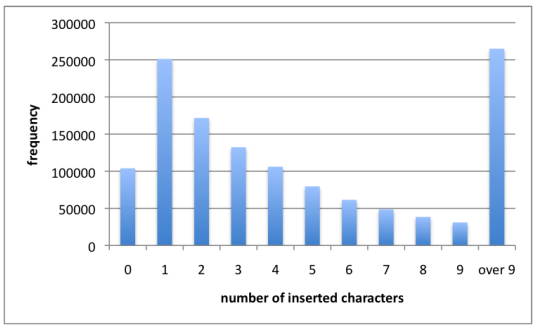

Figure 4: Summary of number of insertion

heuristics to extract corrected sentences from Lang-8. First, we remove all the `<span class="sline">` tags and characters within them. Then, we discard other tags, retaining the characters surrounded by the tags. After this rule, we obtain the corrected sentence shown in the bottom row in Table 4.

2.4 Data Statistic and Filtering by Edit Distance

In an actual correction, it is expected that annotators do not completely rewrite the original sentence and most character strings remain the same as the original sentence. Thus, we investigated the quantitative distribution of Lang-8 data by breaking down the sentences according to the edit distance between the original and corrected sentences (number of deletion / number of insertion of characters in revision log).

Figures 3 and 4 summarize the numbers of deleted and inserted characters. These figures reveal that both distributions of deletion and insertion are comparable. On the other hand, they differ in the absolute frequency of deletion and insertion. For example, the number of cases with no deletion is considerably more than the number with no insertion. Also, the frequency of sentences with more than nine insertions is higher than that for deletions. This reflects the fact that there are many sentences with comments (insertions) and that people tend not to remove too many characters to keep the information of the original sentence written by the learner.

From observations of the created corpus, a correction can be divided into two types: (1) a correction by insertion, deletion, or substitution of strings, (2) a correction with a comment. Table 4 shows examples of correction from Lang-8. The first example is a sentence written by JSL learners containing an error, and is corrected by inserting a character. In the second example the learner's sentence is correct; in addition the annotator writes a comment [11]. Besides, there exist "corrected" sentences to which only the word "GOOD" is appended at the end. In this case, original sentence is not modified at all by the annotator. The inserted comment merely informs the learner that there is no mistake in the learner's writing.

To handle these comments, we conduct the following three pre-processing steps: (1) if the corrected sentence contains only "GOOD" or "OK", we do not include it in the corpus, (2) if edit dis-

[11] Some annotator erase a learner's original sentence and rewrite it to "OK".

Table 3: Extracting corrected sentence from HTML

Sentence written by a JSL learner	去年は参加してなかった、見るだけ。 (I was not participating last year, just watching.)
Corrected sentence with tags	去 年 は 参 加し て な かったせ ず に、見るだけだった。
Seen on the browser	去年は参加~~してなかった~~せずに、見るだけだった。
Corrected sentence	去年は参加せずに、見るだけだった。 (I did not participate but watched last year.)

Table 4: Examples of correction in Lang-8

Sentence written by a JSL learner	ビデオゲームをやました (Video games Yamashita.)
Sentence corrected by an annotator	ビデオゲームをや り ました (I played video games.)
Sentence written by a JSL learner	銭湯に行った。 (I went to a public bath.)
Sentence corrected by an annotator (with comment)	銭湯に行った。 いつ行ったかがある方がいい (I went to a public bath. It is better to say when you went.)

tance between the learner's sentence and corrected sentence is 5, we simply drop the sentence for the corpus, and (3) if the corrected sentence ends with "GOOD" or "OK", we remove it and retain the sentence pair. As a result, we obtained a corpus of 849,894 corrected and aligned sentence pairs by JSL learners.

Another notable issue is that annotators may not correct all the errors in a sentence. Table 5 shows an example of JSL learner's sentence for confusing case markers of "が" (NOM) and "は" (TOP). In this example, "は" and "が" should be corrected to "が" and "は", respectively. However, the annotator left the second case markers "は" unchanged. Because the number of these cases seems low, we regard it as safe to ignore this issue for creating the corpus.

3 Error Correction Using SMT

In this study, we attempt to solve the problem of JSL learners' error correction using the SMT technique. The well-known SMT formulation using the noisy channel model (Brown et al., 1993) is:

$$\hat{e} = \arg\max_{e} P(e|f) = \arg\max_{e} P(e)P(f|e) \quad (1)$$

where e represents target sentences and f represents source sentences. $P(e)$ is the probability of the language model (LM) and $P(f|e)$ is the probability of the translation model (TM). TM is learned from sentence-aligned parallel corpus while LM is learned from target language corpus.

To adapt SMT to error correction, f can be regarded as the sentences written by Japanese learners, whereas e represents the manually-corrected Japanese sentences. TM can be learned from the sentence-aligned learners' corpus. LM can be learned from a monolingual corpus of the language to be learned. Once we obtain a manually-corrected corpus of language learners, it is possible to translate erroneous sentences into correct sentences using SMT.

The use of SMT for spelling and grammar correction has the following three advantages. (1) It does not require expert knowledge. (2) It is straightforward to apply SMT tools to this task. (3) Error correction using SMT can benefit from the improvement of SMT method.

Related work on error correction using phrase-based SMT includes research on English and Japanese (Brockett et al., 2006; Suzuki and Toutanova, 2006). Brockett et al. (2006) proposed to correct mass noun errors using SMT and used 45,000 sentences as training sets randomly extracted from automatically created 346,000 sentences. Our work differs from them in that we (1) do not restrict ourselves to a specific error type such as mass noun; and (2) exploit a large-scale real world data set. Suzuki and Toutanova (2006) proposed a machine learning-based method to pre-

Table 5: Problem of correction in Lang-8

Sentence written by a JSL learner	この４つ が 僕 は 少年のころに発売されて (As for me, these four were sold when I was a kid.)
Sentence corrected by an annotator	この４つ は 僕 は 少年のころに発売されて (As for these four, I was sold when I was a kid.)
Corrt sentence	この４つ は 僕 が 少年のころに発売されて (As for these four, they were sold when I was a kid.)

dict Japanese case particles using a monolingual corpus in the context of SMT.

3.1 Statistical Error Correction with Different Granularity of Tokenization

When translating a sentence from Japanese to another language with SMT, one usually performs word segmentation as a pre-processing step. However, JSL learners' sentences contain a lot of errors and hiragana (phonetic characters), which are hard to tokenize by traditional morphological analyzer trained on newswire text. Suppose we want to tokenize into words the following real sentence written by a JSL learner:

でもじょずじゃりません

The correct counterpart would be:

でもじょう ずじゃあ りません
(But I am not good at it.)

The corrected sentence has "う" and "あ" inserted [12]. These sentences written by a learner and corrected by a native speaker are tokenized as follows by MeCab [13], which is one of the most popular Japanese Morphological Analyzer:

でも じ ょずじゃりません
(but (fragment) (garbled word))

でも じょうず じゃ あり ません
(but good at be not)

These examples illustrate the difficulty of correcting JSL learners' sentence using word-wise SMT.

To alleviate this problem, we propose to build a character-wise segmented corpus with phrase-based SMT. Character-wise model is not affected by word segmentation errors, thus it is expected to be more robust for the task of correcting JSL errors. For the above-mentioned two example sentences, we split sentences into characters rather than words:

This enables the phrase-based SMT to learn the alignment between "じょず" and "じょうず", resulting in a more robust model to correct JSL errors than word-wise model.

4 Experiments on JSL Learner's Error Correction with SMT

We carried out an experiment to see (1) the effect of granularity of tokenization as described in Section 3.1; (2) the effect of corpus size; (3) the difference of L1 model. We used Moses 2.1 [14] as a decoder and GIZA++ 1.0.5 [15] as an alignment tool. We used Japanese morphological analyzer MeCab 0.97 with UniDic 1.3.12 [16] for word segmentation.

We created a word-wise model as baseline. Hereafter, we refer to this as W and also constructed model with entries from UniDic for better alignment, denoted as W+Dic. We used word trigram as LM for W and W+Dic. We built two character-wise models: character 3-gram and 5-gram represented as C3 and C5, respectively. We also conducted minimum error rate training (MERT) (Och, 2003) in all experiments [17].

4.1 Experimental Data

All the data was created from 849,894 Japanese sentences extracted from revision logs of Lang-8 crawled in December 2010. To see the difference of errors stemming from L1, we carried out an experiment with two L1s: English and Mandarin. ALL extracts training data from the entire corpus for the translation model. There are 320,655 Japanese sentences whose L1 is English

[12] It is hard for JSL learners of certain L1 to distinguish Japanese short and long vowels.

[13] http://mecab.sourceforge.net/

[14] http://www.statmt.org/moses/

[15] http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html

[16] http://www.tokuteicorpus.jp/dist/

[17] We performed minimum error rate training to maximize BLEU (5-gram).

and 186,807 Japanese sentences whose L1 is Mandarin. For each L1 JSL corpus, we split the corpus into two parts: 500 sentences for testing and development, and the rest for training.

We shuffled the training data to prepare the corpus for learning LM and TM. We manually reannotated 500 sentences to make gold-standard data and used 200 sentences for testing, and 300 sentences for development.

4.2 Evaluation Metrics

As evaluation metrics, we use automatic evaluation criteria. To be precise, we used recall (R) and precision (P) based on longest common subsequence (LCS) (Mori et al., 1999; Aho, 1990) and character-based BLEU (Papineni et al., 2002).

Park and Levy (2011) adopted character-based BLEU for automatic assessment of ESL errors. We followed their use of BLEU in the error correction task of JSL learners. Since we perform minimum error rate training using BLEU we can directly compare each model's performance.

Recall and precision based on LCS are defined as follows:

$$Recall = \frac{N_{LCS}}{N_{SYSTEM}}, \quad Precision = \frac{N_{LCS}}{N_{CORRECT}}$$

where N_{LCS}, N_{SYSTEM}, and $N_{CORRECT}$ denote the number of character containing longest common subsequences of system results and corrected answers, the number of character containing system results, and the number of character containing corrected answer, respectively. Also, F-measure is the harmonic average between R and P. To illustrate recall and precision based on LCS, let us consider the following example:

CORRECT: 私 は 学生 です
(I am a student)

SYSTEM: 私 わ[18] 学生
(I ring student)

LCS consists of three characters "私 学 生", and $N_{LCS} = 3$. Number of characters in the corrected sentence is six and that in the system is four, so $N_{CORRECT} = 6$ and $N_{SYSTEM} = 4$. Thus, $Recall = 3/4$ and $Precision = 3/6$.

4.3 Experimental Results

Comparison of granularity of tokenization Table 6 illustrates the performance with different methods (Training Corpus: L1 = ALL; Test Corpus: L1 = English; TM: 0.3M sentences; LM: 1M sentence). The character-wise models outperform the word-wise model in both recall and precision. C5 achieved the best precision, F and BLEU, while C3 obtained the best recall.

Table 6: Comparison of the performance (recall, precision, F, BLEU) of error correction for each system with different granularity of tokenization (TM: 0.3M sentences, LM: 1M sentence)

	W	W+Dic	C3	C5
R	0.9043	0.9083	**0.9089**	0.9083
P	0.9175	0.9210	0.9234	**0.9243**
F	0.9109	0.9146	0.9161	**0.9162**
B	0.8072	0.8101	0.8163	**0.8181**

Effects of corpus size We varied the size of TM while fixing the size of LM to 1M sentences to see the effect of corpus size on the performance. Figure 5 shows the performance (BLEU) with different TM size (Training Corpus: L1 = ALL; Test Corpus: L1 = English). The larger the size of the TM, the higher the BLEU. This confirms that the large scale JSL learner's corpus extracted from Lang-8 is a great source of learning learners' errors. Although TM trained on 0.85M sentences exhibits lower performance than TM trained on 0.3M sentences, the difference is not statistically significant.

Comparison of L1 of the training model Table 7 shows result for each L1 trained translation model [19].

Basically, performance was better when TM was trained with the same L1 as the test set. In the case where L1 of test data is English, the model trained from ALL is comparable to L1 English. This is because the model trained from ALL includes many sentence written by learners whose L1 is English.

5 Discussion

As we discussed in Section 2, the extracted corpus still contains comments in the corrected sentences. However, it does not greatly affect the performance of the JSL learner's error correction, demonstrating that we were able to build a large-scale JSL learners' corpus from revision logs. Moreover, we have checked all the output of

[18] The pronunciation of "わ" is the same as "は".

[19] Note that LM was trained from the whole training corpus. We did not change L1 for LM.

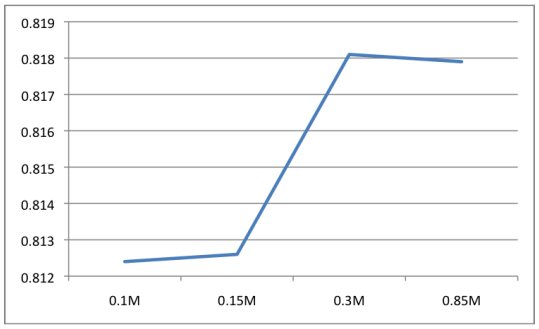

Figure 5: Comparison of the performance (BLEU) of error correction for different size of TM (fixing the size of LM to 1M sentence)

our SMT-based error correction system, but none of the errors of the system derive from the annotators' comments.

Here are some examples illustrating the difference of the scale of the training corpus.. We compared translation models TM trained on 0.1M sentences and 0.3M sentences in Figure 5. Note that the model trained on 0.1M sentences gave the worst result, wheares model trained on 0.3M sentences achieved the best. All the models were trained on 1M sentence for LM. Both models corrected the examples below:

Original: またど もう ありがとう
(Thanks, Matadomou (OOV))

Correct: またど うも ありがとう
(Thank you again)

Also, both of them corrected a case marker error frequently found in JSL learners' writing as in:

Original：TRUTHわ 美しいです
(TRUTH wa beautiful)

Correct：TRUTHは 美しいです
(TRUTH is beatiful)

On the other hand, the model trained on 0.3M sentences corrected the following example:

Original: 学生な る たら学校に行ける
(the learner made an error in conjugation form.)

Correct: 学生な った ら学校に行ける
(Becoming a student, I can go to school.)

0.1M: 学生な るため 学校に行ける
(I can go to school to be student)

0.3M: 学生な った ら学校に行ける
(Becoming a student, I go to a school)

Table 7: Comparison of the performance (recall, precision, F, BLEU) of error correction trained on different first language (L1). (TM: 0.18M sentences, LM: 1M sentences)

			L1 of test data	
			English	Mandarin
L1 of training data	English	R	0.9079	0.9339
		P	0.9241	0.9387
		F	0.9159	0.9363
		B	**0.8148**	0.8573
	Mandarin	R	0.9063	0.9357
		P	0.9169	0.9388
		F	0.9116	0.9373
		B	0.8083	**0.8589**
	ALL	R	0.9099	0.9349
		P	0.9183	0.9367
		F	0.9141	0.9358
		B	0.8121	0.8553

This example also illustrates the fact that there remains uncorrected errors (missing "ni" case marker after "学生" *student*) as we discussed in Section 2.4.

Another remaining issue is evaluation metric. We have used character-based BLEU, recall and precision based on the longest common subsequence. These methods have the advantage of allowing automatic system evaluation, but they do not reflect the importance of the errors that language learners make. There is still much room for improvement in the evaluation metric for error correction of language learners.

6 Conclusions

We proposed to extract a large-scale learners' corpus from the revision log of a language learning SNS. This corpus is easy to obtain in a large-scale, covers a wide variety of topics and styles, and can be a great source of knowledge for both language learners and instructors. We adopted phrase-based SMT approaches to alleviate the problem of erroneous input from language learners. Experimental results show that the character-wise model outperforms the word-wise model. We plan to apply factored language and translation models incorporating the POS information of the words on the target side, while learners' input is processed by a character-wise model.

Acknowledgements

We would like to thank Hiromi Oyama, Seiji Kasahara and Joseph Irwin for their useful comments on this study. Special thanks to Yangyang Xi for maintaining Lang-8.

References

Alfred V. Aho. 1990. Algorithms for Finding Patterns in Strings. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science: Volume A: Algorithms and Complexity*, pages 255–300. Elsevier, Amsterdam.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of ACL*, pages 50–57.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL Errors Using Phrasal SMT Techniques. In *Proceedings of COLING-ACL*, pages 249–256.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):266–311.

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 japanese). In *Proceedings of the Information Processing Society of Japan SIG*, pages 39–46.

Xiaohua Liu, Bo Han, and Min Zhou. 2011. Correcting Verb Selection Errors for ESL with the Perceptron. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part II*, CICLing'11, pages 411–423, Berlin, Heidelberg. Springer-Verlag.

Shinsuke Mori, Masatoshi Tsuchiya, Osamu Yamako, and Makoto Nagao. 1999. Kana-Kanji Conversion by a Stochastic Model (in japanese). *Transaction of Information Processing Society of Japan*, 40(7):2946–2953.

Ryota Nampo, Hokuto Ototake, and Kenji Araki. 2007. Automatic Error Detection and Correction of Japanese Particles Using Features within Bunsetsu (in japanese). In *Proceedings of the Information Processing Society of Japan SIG*, pages 107–112.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL*, pages 160–167.

Hiromi Oyama and Yuji Matsumoto. 2010. Automatic Error Detection Method for Japanese Case Particles in Japanese Language Learners. In *Corpus, ICT, and Language Education*, pages 235–245.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, pages 311–318.

Y. Albert Park and Roger Levy. 2011. Automated Whole Sentence Grammar Correction Using a Noisy Channel Model. In *Proceedings of ACL*, pages 934–944.

Alla Rozovskaya and Dan Roth. 2011. Algorithm Selection and Model Adaptation for ESL Correction Tasks. In *Proceedings of ACL*, pages 924–933.

Hisami Suzuki and Kristina Toutanova. 2006. Learning to Predict Case Markers in Japanese. In *Proceedings of ACL*, pages 1049–1056.

Huichao Xue and Rebecca Hwa. 2010. Syntax-Driven Machine Translation as a Model of ESL Revision. In *Proceedings of COLING*, pages 1373–1381.