International Journal of Technology (2017) 3: 549-558
ISSN 2086-9614 © IJTech 2017

GENERATING ARTIFICIAL ERROR DATA FOR INDONESIAN PREPOSITION ERROR CORRECTIONS

Budi Irmawati[1,2*], Hiroyuki Shindo[1], Yuji Matsumoto[1]

[1]*Computational Linguistics laboratory, Nara Institute of Science and Technology*
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
[2]*Informatics Engineering Department, University of Mataram*
Jl. Majapahit 62 Mataram 83125, NTB, Indonesia

(Received: July 2016 / Revised: April 2017 / Accepted: April 2017)

ABSTRACT

Large-scale annotated data written by second language learners are not always available for low-resource languages such as Indonesian. To cope with data scarcity, it is important to generate 'learner-like' artificial error sentences when the available real learner data is insufficient and language experts cannot construct data. In this paper, we propose a new method for generating effective error-injected artificial data to proliferate training examples for preposition error correction tasks. Our method first generates a large scale of noisy artificial error data via the use of a simple error injection method. It then selectively removes the uninformative (noisy) instances from the artificial data. We assume that 'good' artificial preposition error data would be effective training data for error correction tasks. Therefore, to evaluate the goodness of the generated artificial data, we used the generated artificial data as training data to correct preposition errors in real learners' sentences. The results of our study indicate that the use of our artificial data for training improves preposition error correction performance. The results also show that training on a smaller sized of good instances outperforms training on much larger-sized noisy instances as well as that on sentences written by native speakers. This method is language-independent and easy to apply to other low-resource languages because it assumes only a small size of learner error data and uses features that could be extracted automatically from linguistically annotated sentences.

Keywords: Artificial data; Indonesian language; Low-resourced languages; Noise removal; Preposition error correction

1. INTRODUCTION

The number of research works conducted on the error correction of second language (L2) learners' text, particularly on resource-rich languages such as English, Chinese, and Japanese, are increasing. On the other hand, for low-resource languages, the scarcity problem persists: Learners' text is not always available publicly. The well-known problems in building a language learner corpus are as follows: *First*, learner corpora are not always available in an electronic format; most are handwritten on paper and require human effort to be transcribed into a machine-readable format. *Second*, the learner corpora, if available, are not annotated with error tags. *Third*, the learner corpora also need to be annotated with linguistic information, such as part-of-speech (PoS) and syntactic information.

*Corresponding author's email: budi-i@is.naist.jp, Tel. +62-370-636126, Fax. +62-370-636523
Permalink/DOI: https://doi.org/10.14716/ijtech.v8i3.4825

In previous research (Cahill et al., 2013; Han et al., 2010), preposition error correction systems trained on annotated error sentences from learner corpora outperformed the systems trained on larger sizes of well-written native sentences when the systems were used to correct preposition errors in real learner data. The performance of the systems trained on sentences with errors improved because the systems learned error patterns from the annotated corpora. Our preliminary investigation of applying a similar model to learner data (in cross-validation) showed the same tendency. However, the performance level was not significantly high because of the lack of large-scale training data.

Inserting artificial errors into well-written sentences (Foster & Andersen, 2009; Izumi et al., 2003; Wagner et al., 2009) is an alternative method of coping with a limited size of learner data. Izumi et al. (2003) proposed a method for replacing a correct article with three other articles or an empty article. Wagner et al. (2009) created grammatical and non-grammatical errors, such as missing words, extra words, verb forms, agreement, and word spelling. Foster and Andersen (2009) developed an artificial error tool, GenERRate, to automatically insert syntactic errors by substituting, removing, inserting, or relocating a correct word within a sentence. Unfortunately, their results showed that accuracy drops when their system was trained on sentences with artificial errors but tested on real learner data. In reality, learners do not make mistakes randomly: Rather, they make mistakes based on the conceptualization in their first language (L1) (Leacock et al., 2014). With respect to Indonesian, for example, the preposition *in* as 'in the room' may be translated as *di* ('in') or *di dalam* ('inside') based on the context. A good explanation of the choice of a preposition was provided by Leacock et al. (2014) in the example 'driving at a high speed,' which is translated in Indonesian as *mengendarai pada kecepatan tinggi*. In Indonesian, the appropriate preposition choice should be *dengan* ('with') instead of *pada* ('at'). Our use of the Google search engine resulted in 3,050 hits for *"mengendarai dengan kecepatan tinggi"* but only in 342 hits for *"mengendarai pada kecepatan tinggi."*

To generate better artificial data, Rozovskaya and Roth (2010) proposed a new artificial error-generation method by selectively injecting errors into native sentences based on error distributions in learners' sentences and by using L1 information. This method was successfully implemented in grammatical error detection and in the correction of articles and prepositions in sentences written by learners of English whose L1 languages were Chinese, Russian, and Czech. When the authors applied this method to small-sized learner data written by Czech speakers, the outcome showed that randomly injecting errors to training data (without L1 distribution information) resulted in a better performance, indicating that the statistic of small-sized data was less reliable. Similar to the Czech case, we were unable to apply the method proposed by Rozovskaya and Roth (2010) because the small size of our learner data did not allow us to obtain a reliable distribution of the learners' L1.

Without error distribution information, one may generate artificial error data by first injecting errors randomly and then improving the quality of the artificial data by removing instances that involve noise. However, the removal of noisy instances may also remove informative instances if the gold-standard data are not sufficient to build an accurate predictor for identifying noisy instances (Martineau et al., 2014). Therefore, the removal of noisy instances has a significant effect.

We propose a new method for generating effective artificial error data that resemble learner data. Our goal is to proliferate learner data when the real learner data is insufficient and difficult to enlarge. This method calculates a score for each artificial instance produced by a simple error injection method while using minimal resources. In addition, we also exploit a linguistic feature such as *clitics*, which are attached to prepositions, verbs, or nouns, to obtain the base form of those word types. Our proposed method includes three steps: (1) error injection; (2) scoring;

and (3) selection. The *error injection* step injects preposition errors into well-written native sentences, the *scoring* step assigns a score to each artificial error instance by estimating its informativeness, and the *selection* step removes all artificial error instances that are considered to be uninformative.

We first applied our proposed method to preposition errors because, in the case of Indonesian, preposition error correction tasks have not been thoroughly explored. Even though we lacked learner data (in Indonesian), as the confusion set[1] of preposition errors is small compared to other error types, we obtained sufficient raw samples for each preposition error. Moreover, in the case of English, preposition choice is still challenging for L2 learners (Chodorow et al., 2007).

We argue that our proposed method generates effective artificial preposition error data when the size of available learner data is small. By training various data in an error correction task, we show that randomly error-injected native data perform better than the original native data. We then prove that artificial error data generated using our method (by selectively removing noisy instances from the error-injected data) outperform native/randomly-error-injected data. Then, to evaluate whether our method produces 'good' artificial preposition error data, we train a preposition error correction model on the generated artificial error data to see their contribution to correcting preposition errors in real learners' sentences.

The paper is organized as follows: Section 2 describes our method for generating artificial error data, and Section 3 explains the experiments. In Section 4, we describe the results and present a discussion. Section 5 concludes with a summation of the importance of this work and suggestions for future directions.

2. THE ARTIFICIAL ERROR GENERATION METHOD

In this section, we present our proposed *artificial error generation method*. Our method includes three steps, as shown in Figure 1. Details of each step are explained in Subsections 2.1 and 2.3. We also explain two components used in our method: *confusion set* and *preposition error correction model* (PECM) in Subsections 2.4 and 2.5. The former is used in the error injection step while the latter is use in the scoring and selection steps.

Figure 1 The steps in artificial error generation

2.1. Error Injection

The error injection step replaces a correct preposition in native sentences with an incorrect preposition from the confusion set (explained in Subsection 2.4). For example, the words *oleh, di, untuk, kepada, and pada* are those that are confused with *bagi*, so we replaced the preposition *bagi* in a sentence with all prepositions in the confusion set to generate artificial error sentences (see Figure 2).

2.2. Scoring

This step calculates a *fitness score s* to measure the usefulness of an instance in the artificial error data. As we assume that 'good/informative' artificial training data for the PECM predicts better corrections, this step uses the PECM to calculate the fitness score for each artificial error

[1] *Confusion set* is a set of an incorrect word and all its possible corrections as given by native speakers. See Subsection 2.4 for detailed explanations.

instance. The PECM calculates an F_1 score when it predicts corrections on real learner data containing preposition errors.

original native sentence:	*UMY memberikan beasiswa*	*bagi*	*mahasiswa Korea*
artificial error sentence:	*UMY memberikan beasiswa*	*oleh* *di* *untuk* *kepada* *pada*	*mahasiswa Korea*

Figure 2 Error injection based on the confusion set. The preposition *bagi* is replaced with other prepositions in the confusion set (*oleh, di, untuk, kepada,* and *pada*)

The calculation of the fitness score is illustrated in Figure 3. We assume that if an informative instance is removed from the training data, the F_1 score decreases. On the contrary, if an uninformative instance is removed from the training data, the F_1 score increases. Suppose the following: *A* is a set taken from the artificial error data produced by the injection step, *e* is an observed instance taken from *A*, and *L* is our learner data, which are used as the PECM test data. As shown in Figure 3, this step has two parts. The *first* part calculates $F_1(A)$ as an F_1 score of PECM trained on the whole *A* and tested on *L*. The *second* part calculates the fitness score *s* for each *e*, denoted as s_e. The calculation of s_e is described in the following steps:

(1) Remove *e* from *A*. The set remaining after *e* is removed from *A* is called *A-e*.

(2) Calculate $F_1(A\text{-}e)$ as the F_1 score of PECM trained on *A-e* and tested on *L*.

(3) The fitness score *s* of *e*, s_e, is calculated as the difference between $F_1(A)$ and $F_1(A\text{-}e)$.

We repeat steps (1) to (3) for each *e* in *A*.

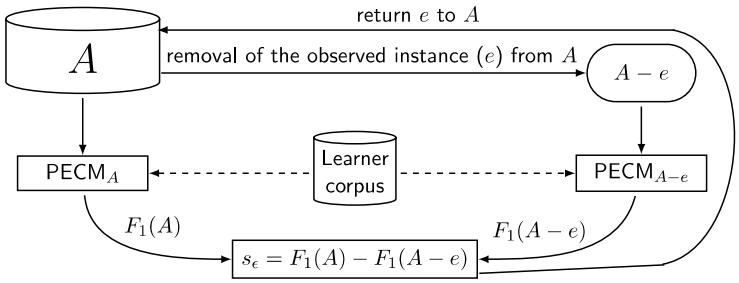

Figure 3 Calculation of the fitness score *s* of each instance *e*. *A* is a set of native sentences injected with preposition errors; we removed *e* from *A* and called the remaining set *A-e*; PECM$_A$ is PECM trained on *A*; PECM$_{A\text{-}e}$ is PECM trained on *A-e*. $F_1(A)$ and $F_1(A\text{-}e)$ are the F_1 scores of PECM$_A$ and PECM$_{A\text{-}e}$ as tested on the real learner sentences

2.3. Selection

This step also consists of two parts. The first part removes artificial instances whose *s* score is lower or equal to zero. The instances whose *s* score is lower than zero decrease the F_1 score of PECM; the instances whose *s* score is equal to zero do not affect the PECM performance. We call the remaining artificial instances $\tilde{A}$. The second step sorts the instances by s_e scores of $\tilde{A}$. It then sets a threshold *a* so that PECM trained on $\tilde{A}$ and tested on *L* (*L* is the same as that used in the scoring step) obtains a maximum F_1 score as expressed in Eq. 1. $\tilde{A}_{s>a}$ is the subset of $\tilde{A}$, whose *s* is higher than *a*.

$$\hat{a} = \underset{a}{\operatorname{argmax}} F_1(\tilde{A}_{s>a}|L) \quad (1)$$

The optimal value of *a* can be found effectively by performing a binary search on the sorted list

of $\tilde{A}$.

2.4. Confusion Set

For each original preposition p_o written by a learner in the learner data that is incorrect, there is a correction p_c given by a native speaker. A set of an ordered pair (p_c, p_o) is called a *confusion set*. So, for each preposition p_c that occurs in the learner data, we extract all possible p_o and obtain about 100 (p_c, p_o) pairs. The number of confusion prepositions for each preposition varies in the range of 1 to 12, with an average 4.17. Then, as we work on only 11 prepositions, the recall of the confusion set is 0.651.

2.5. Preposition Error Correction Model (PECM)

From 21 incorrect prepositions in our learner data, our PECM covered the first 11 prepositions with error frequencies above five to obtain more error variations. To build the model, we trained naïve Bayes classifiers because the training time of this technique is short, and it is efficient in learning parameters for a classification task.

To conduct a multi-class classification, we used the *one-vs-all* approach. For each preposition p, we set feature vectors of p as positive samples and feature vectors of other *M-1* prepositions as negative samples, where *M* is the number of the target prepositions. Then, to identify the predicted preposition that fits the sentence, we first filtered them out based on the confusion set and ranked the candidate corrections based on their confidence scores as calculated by the classifier.

Table 1 lists feature types used to build PECM, which consists of 37 features. Based on this table, we extract features such as *n*-gram, morphological information, and syntactic (dependency) information using the language resources explained in Subsection 3.1. The table shows some feature formats followed by their examples in the right column of the table. The examples are derived from the example sentence in Figure 4.

Table 1 Examples of features used to train PECM

Feature Types	Feature formats	Examples
word *n*-grams	$w_{-1}\|w_0$, $w_0\|w_1\|w_2$	*beasiswa|di, di|mahasiswa*
PoS *n*-grams	$p_{-1}\|p_0$, $p_0\|p_1\|p_2$	VERB|NOUN|NOUN, NOUN|NOUN
dependencies	$hw_0\|hwRel_0$, $hw_0\|chRel_1$, $hw_0\|hPos_0$	*memberikan|mahasiswa, memberikan|prep, memberikan|pobj, memberikan|left*
PoS dependencies	$hp_0\|chp_0$	TRAN|NOUN
affixes	$hPre_0\|hSu_0$	*meN|kan*

Note. The examples in the right column are derived from the learner sentence, "UMY memberikan beasiswa di mahasiswa Korea." The incorrect preposition *di* is considered the target word *w0*. The dependency relations of that sentence are drawn in Figure 4.

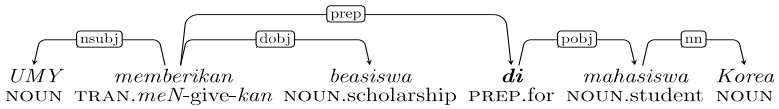

Figure 4 An example of an artificial sentence annotated with dependency relations

3. EXPERIMENTAL METHOD

The goal of the experiment is to evaluate whether our method generates 'good' artificial error sentences. For that, we trained the PECM on various data (explained in Subsection 3.3) and

tested it on the learner data. We then compared the F_1 scores of the PECM trained on all of those data.

3.1. Language Tools and Resources

3.1.1. Part-of-speech tagger

We used Morphind,[2] a morphological analysis system for Indonesian (Larasati et al., 2011) that covers affixations and clitics, to PoS tag all sentences in our data.

3.1.2. Dependency parser

We took 1,132 short corrected-learners' sentences from learner data and split them into two sets: 1,032 sentences as training data and 100 sentences as test data. We used short sentences only (with as many as 20 words), because shorter sentences are sufficient for obtaining good accuracy for parsing dependency relations related to prepositions. We then manually annotated the sentences with labeled dependency relations and trained the Minimum Spanning Tree (MST) parser (McDonald et al., 2006) on the training data. Our parser achieved an accuracy of 81.2% when evaluated on the test data.

3.1.3. Native data as the source sentences for the artificial error data

We used a newspaper dataset from the Indonesian part of Leipzig corpora (Quasthoff et al., 2006) that contained 1M sentences with 1.5M prepositions. From the corpora, we took only the short native sentences containing at least one preposition to achieve better dependency relationship accuracy.

3.1.4. Learner data

We used the aligned Indonesian learner errors and corrected sentences from the Lang-8[3] Website that were crawled in 2011 (Mizumoto et al., 2011). The learner data contains 896 journals that are extracted to 6,488 pairs of learners' sentences and 77,201 tokens (Irmawati et al., 2016a).

We extracted all learners' sentences containing at least one preposition and asked a native speaker, who holds a master degree in social science, to check the alignment and to filter out inappropriate error sentences. We considered an error sentence inappropriate if it was semantically incomprehensible. A preposition selected by learners was correct if it was acceptable by the context even though the Indonesian Lang-8 users who corrected it replaced the correct preposition with another correct preposition.

We obtained 5,502 prepositions from sentence pairs in which 382 prepositions were replacement errors (about 6.94% of all preposition usages). However, only 297 preposition errors were corrected (replaced) with another preposition. We have also used these data to generate artificial error data with a different method (Irmawati et al., 2016b).The remaining preposition errors were corrected to words other than prepositions. In the experiments, we used the 297 replacement preposition errors as training data for the artificial error generation and as test data for PECM (to evaluate our method) in a binary cross-validation.

3.2. Pre-Processing

In Indonesian, clitics may attach to verbs, nouns, or prepositions. Therefore, to obtain a preposition, its head, and its object correctly, we split off a clitic from the attached word. For instance, for the word *dengannya* ('with him/her'), we split the clitic *–nya* ('him/her') from the preposition *dengan* and tagged *–nya* as a pronoun. Next, we assigned the preposition as the head of *–nya* and labeled *–nya* as a preposition object (pobj).

To identify word types and to enable the extraction of affixes (and clitics) in sentences, we ran

2 http://septinalarasati.com/work/morphind/

3 http://lang-8.com

Morphind and then changed the clitics back to their original forms. We also normalized numbers in the sentences, deleted unnecessary punctuation, and removed hyphens. We then used the trained MST parser to assign dependency relations to the sentences and extracted the syntactic information (dependency and PoS dependency), as listed in Table 1.

3.3. Experiments

We randomly split the learner data (297 instances) into two subsets with similar preposition error distribution: the training set and the test set. The training set was used as training data to generate the artificial error data in the *scoring* and *selection* steps. The test set was used by the PECM as test data to evaluate the generated artificial error data. We used the two subsets in a binary cross-validation and reported the average score.

To analyze the results, we used five training data sets as follows:

(1)LEARNER: This dataset uses all learner data for training and testing PECM in a 10-fold cross-validation.

(2)CLEANNATIVE: This data set uses the native sentences as training data. However, because native data do not contain preposition errors (prepositions selected by learners), we excluded all features that contain w_0 (see Table 1) from the feature vector of each targeted preposition instances.

(3)RNDINJECT: This data set uses the native data injected randomly with another preposition.

(4)CSINJECT: This data set uses the native data injected with all possible preposition errors, based on the confusion set.

(5)**ArtScoring**: This data set uses the artificial sentences obtained from our proposed method.

We considered CLEANNATIVE, RNDINJECT, and CSINJECT as the baselines for our experiments, while LEARNER was used as a reference because its source data was different from other training data. We reported the experiment results on precision and recall (Dahlmeier & Ng, 2011) because we were concerned only with how well PECM can correct preposition errors in the learner data.

4. RESULTS AND DISCUSSION

The experiment was tested on real learner data. Figure 5 shows the comparison of PECM trained in five cases. As a reference, the magnified graph in Figure 5 illustrates that LEARNER outperforms other data sets when the size of the training data is as small as 300 instances. The main graph shows that CLEANNATIVE obtains the lowest results and cannot outperform LEARNER even when the size of the training data is increased to 50K. It also shows that **all artificial data sets** perform better than both CLEANNATIVE and LEARNER when their size is larger. Note that CSINJECT obtains its highest F_1 score at 0.623 when its size is 270K, while **ArtScoring** obtains a 0.667 F_1 score using only 90K training instances. In considering up to a 90K training size, other baselines do not outperform **ArtScoring**.

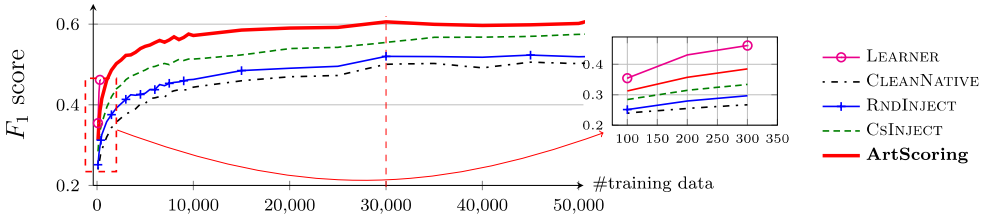

Figure 5 PECM results tested on real learner data (**ArtScoring** is the artificial training data obtained from our proposed method). It shows that **ArtScoring,** compared to other artificial data, uses less

training data to perform similar results. The magnified graph shows that LEARNER is superior when the size of training data is small.

Figure 5 indicates that among artificial training data, a lesser amount of training data generated using our method (**ArtScoring**) outperforms all baselines, leading to the supposition that selectively removing noise from artificial error data indeed results in better performance. Using only 10K instances as examples, **ArtScoring** significantly improves the error correction performance by approximately 0.16 points compared to CLEANNATIVE (calculated by a two-tailed test with $p < 0.0001$). Comparing **ArtScoring** and RNDINJECT/ CSINJECT, we can see that the former achieves the same performance as the other two with a much smaller size of training data. This figure also shows that **ArtScoring,** compared to the three baselines, requires less training data to obtain a similar performance, because our method removes uninformative instances. However, **ArtScoring** requires 1K training instances to obtain a performance similar to that of LEARNER, meaning that our method requires improvement.

Table 2 shows a detailed comparison of precisions, recalls, and F_1 scores of PECMs trained on 30K instances, in which the results are maximum, based on Figure 5.

Table 2 PECM results of data trained on 30K instances in which the results are maximum are presented in Figure 5

Training data	Size	Precision	Recall	F_1
CLEANNATIVE	*30K*	0.698	0.389	0.500
RNDINJECT	*30K*	0.707	0.411	0.520
CSINJECT	*30K*	0.716	0.452	0.554
ArtScoring	*30K*	**0.767**	**0.500**	**0.606**
LEARNER (reference)	*300*	*0.633*	*0.368*	*0.462*

Table 3 Confusion matrix resulted by PECM trained on **ArtScoring** and tested on the real learner data

Gold Answers	Predictions by PECM										
	dalam	di	dengan	pada	bagi	dari	untuk	ke	kepada	secara	Selama
dalam	**47**	0	4	3	0	4	1	0	0	0	0
di	1	**31**	0	2	0	2	0	0	1	0	0
dengan	1	3	**13**	0	2	0	0	1	0	0	0
pada	2	0	1	**15**	0	0	0	0	1	0	0
bagi	0	1	3	0	**11**	2	1	0	0	0	0
dari	0	2	0	1	1	**9**	0	3	0	1	0
untuk	0	0	2	1	1	0	**10**	0	1	0	0
ke	0	0	1	0	1	0	0	**9**	0	0	0
kepada	*0*	*0*	*1*	*0*	*0*	*2*	*2*	*0*	***2***	*0*	*0*
secara	0	0	0	0	0	1	0	0	0	**6**	0
selama	0	1	0	0	0	1	0	0	0	0	**4**

The reference (LEARNER) and all artificial data support the conclusion put forward by Cahill et al. (2013) that training on error-annotated data outperforms training on well-written data in terms of both precision and recall. RNDINJECT obtains a bit lower performance compared to other artificial error data because it contains more unnatural errors (based on the contexts) and may over-correct errors. The results show that our method improves both precision and recall

among two other artificial training data and that the F_1 score is significantly improved (by 0.052 points) compared to the original artificial error data CSINJECT.
To evaluate the performance of each observed preposition, Table 3 shows the confusion matrix of predictions provided by PECM trained on **ArtScoring**. It shows that PECM highly confuses predicted corrections of *kepada* (meaning 'for' when it is used in a formal context) to *dari* and *untuk*. The former has the opposite meaning; the latter has a similar meaning. Moreover, *kepada* (presented in italics) has an error frequency of only seven in the learner data. On the other hand, some prepositions, such as *dengan* and *dari*, have high levels of confusion with particular prepositions, because both prepositions may be used with wide-coverage verbs. Future exploration of more specific features to distinguish those prepositions is needed.

We noticed a drawback in which our method relied on the learner data that we used to train the artificial error data generation, meaning that we have to train our method on different learner data if we want to perform tasks in different domains. Therefore, we hope to obtain more annotated learner data to work further on NLP tasks related to language learners.

5. CONCLUSION

In this paper, we propose a method for generating effective artificial error data to proliferate learner data when the real learner data is insufficient and difficult to enlarge. We applied a selection process on the native data injected by artificial errors to remove noisy instances. By removing the noisy instances from the training data, we obtained better performance by using smaller-sized training data rather than all the artificial error instances. The experimental results indicated that our proposed method outperforms other error injection methods. This method is useful when information about error distribution is not known and no information is provided about learners' proficiency levels. In addition, the method is easily applied to other low-resourced languages because it assumes only a small size of learner error data. In the future, adding similarity information to this method may be useful for improving the generated artificial error data, for instance, to cope with unseen syntactic context. Artificial data could then be used to generate multiple-choice questions for second language learners. However, obtaining more real learner data is crucial.

6. ACKNOWLEDGEMENT

We acknowledge the Lang-8 users who wrote and corrected the journals. We would like also to thank unknown reviewers, Lis Kanashiro, Kevin Cheng, and Sony H. Wijaya for valuable discussion and comments. This study is supported in part by the DGHE, Republic of Indonesia under BPPLN Scholarship Batch 7 fiscal year 2012-2015.

7. REFERENCES

Cahill, A., Madnani, N., Tetreault, J., Napolitano, D., 2013. Robust Systems for Preposition Error Correction using Wikipedia Revisions. *In:* Proceedings of the Conference of the NACCL: HLT, Atlanta, Georgia, 21st–23rd June 2013, ACL

Chodorow M., Tetreault, J.R., Han, N., 2007. Detection of Grammatical Errors Involving Prepositions. *In:* Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions, Stroudsburg, Pennsylvania, USA, 28th June 2007, ACL, pp. 25–30

Dahlmeier, D., Ng, H.T., 2011. Grammatical Error Correction with Alternating Structure Optimization. *In*: Proceedings of the 49th Annual Meeting of the ACL: HLT - Volume 1, Stroudsburg, Pennsylvania, USA, 19th-24th June 2011, ACL, pp. 915–923

Foster, J., Andersen, Ø.E., 2009. GenERRate: Generating Errors for Use in Grammatical Error Detection. *In:* Proceedings of the 4th Workshop on Innovative Use of NLP for Building

Educational Applications, Pennsylvania, USA, 5th June 2009, Stroudsburg, ACL, pp. 82–90

Han, N., Tetreault, J., Lee, S., Ha, J., 2010. Using an Error-annotated Learner Corpus to Develop an ASL/AFL Error Correction System. *In:* Proceedings of the 7th International Conference on LRE, Valletta, Malta, 23rd May 2010, ELRA, pp. 763–770

Irmawati, B., Komachi, M., Matsumoto, Y., 2016a. Towards Construction of an Error-Corrected Corpus of Indonesian Second Language Learners. *In*: Almeida, F.A. et al. Ed. Input a Word, Analyse the World: Selected Approaches to Corpus Linguistics. Cambridge Scholars Publishing: Newcastle, USA, pp. 425–443

Irmawati, B., Shindo, H., Matsumoto, Y., 2016b. Exploiting Syntactic Similarities for Preposition Error Correction on Indonesian. *In*: Proceedings of The 5th Workshop on Spoken Language Technologies for Under-resource languages, Jogjakarta, Indonesia, 9th-12th May 2016, Procedia Computer Science Volume 81 - Elsevier. pp. 214–220

Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T., Isahara, H., 2003. Automatic Error Detection in the Japanese Learners' English Spoken Data. *In:* Proceedings of the 41st Annual Meeting on ACL - Volume 2, Sapporo, Japan, 7th-12th July 2003, ACL, pp. 145–148

Larasati, S.D., Kuboň, V., Zeman, D., 2011. Indonesian Morphology Tool (MorphInd): Towards an Indonesian Corpus. *In:* Proceedings of the 2nd International Workshop Systems and Frameworks for Computational Morphology, Zurich, Switzerland, 26th August 2011, pp. 119–129

Leacock, C., Chodorow, M. Gamon, M., Tetreault, J., 2014. *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers: Seattle, Washington, San Rafael, California, USA

Martineau, J., Chen, L., Cheng, D., Sheth, A., 2014. Active Learning with Efficient Feature Weighting Methods for Improving Data Quality and Classification Accuracy. *In:* Proceedings of the 52nd Annual Meeting of the ACL (Volume 1: Long Papers), Baltimore, Maryland, USA, 23rd-25th June 2014, ACL, pp. 1104–1112

McDonald, R., Lerman, K., Pereira, F., 2006. Multilingual Dependency Analysis with a Two-stage Discriminative Parser. *In:* Proceedings of the 10th CoNLL, Stroudsburg, Pennsylvania, USA, 8th–9th June 2006, ACL, pp. 216–220

Mizumoto, T., Komachi, M., Nagata, M., Matsumoto, Y., 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. *In:* Proceedings of the 5th IJCNLP, Chiang Mai, Thailand, 8th–11th November 2011, AFNLP, pp. 147–155

Quasthoff, U., Richter, M., Biemann, C., 2006. Corpus Portal for Search in Monolingual Corpora. *In:* Proceedings of the 5th LREC, Genoa, Italy, 24th–26th May 2006, pp. 1799–1802

Rozovskaya, A., Roth, D., 2010. Generating Confusion Sets for Context-sensitive Error Correction. *In:* Proceedings of the 2010 Conference on EMNLP, Stroudsburg, Pennsylvania, USA, 9th–11th October 2010, ACL, pp. 961–970

Wagner, J., Foster, J., Genabith. J., 2009. Judging Grammaticality: Experiments in Sentence Classification. *CALICO Journal*, Volume 26(3), pp. 474–490