Automatic Extraction and Prediction of Word Order Errors From Language Learning SNS

Brendan Flanagan
九州大学大学院 システム情報科学府
b.flanagan.885@s.kyushu-u.ac.jp

廣川 佐千男
九州大学 情報基盤研究開発センター
hirokawa@cc.kyushu-u.ac.jp

1 Introduction

In recent years, research into writing tools to support foreign language learners of English has been growing. However, most research to date has focused on the prediction/correction of prevalent errors in learner writing, such as: preposition and article errors[1]. While the prediction/correction of these errors would have a great impact on learner writing, other less prevalent errors, such as word order errors, have received little attention. Word order differs significantly across languages[2], which poses a particular problem for learners from L1 languages that have a fairly different word order to the L2 language. In this paper, we examine the prediction of word order errors in foreign language writing of learners from a Japanese L1 background learning English. In previous work, the authors have examined automated error prediction of 15 different error categories in learner writing on the language learning SNS, Lang-8.com[3, 4]. However the samples of manually tagged sentences available for some error categories was minimal, such as word order errors, and was problematic when training error models and resulted in low prediction performance. To overcome this problem, we propose that a large amount of word order error samples can be automatically extracted from a corpus of corrected learner writing by comparing the edit distance between original and corrected sentences. We then train and evaluate the prediction performance of a Support Vector Machine (SVM) classifier by analyzing a corpus constructed using the proposed method.

The method of analyzing the edit distance between original and corrected learner writing sentences has been examined in previous work to automatically identify errors[5] and extract L2 criterial lexicogrammatical features from learner corpora[6]. We extend the use of this method to data that has been collected from an language learning SNS to automatically predict word order errors by machine learning.

2 Automatic Word Order Error Sample Extraction by Edit Distance

In this section, we will introduce a method of automatically extracting sentences written by foreign language learners that contain word order errors from a corrected language learning writing corpus. An edit distance of the difference between the original and corrected sentence can be analyzed to identify the corrections that have been made. In particular, we analyzed the Levenshtein distance[7] to find insertions and deletions in corrected sentence pairs.

A word order error can be thought of as a sentence pair that contains the same frequency of insertions and deletions identified by the edit distance for each corrected word. Conversely, a sentence pair that only contains either insertions or deletions for each corrected word can be thought as not containing a word order error. In Equation 1, we define the conditions used to select a set of sentence pairs that contains word order errors.

$$WO(S) = \{s_i | w_j \in s_i; ins(w_j) = del(w_j), ins(w_j) > 0\} \tag{1}$$

Where S is the set of all sentence pairs, w_j is the j^{th} word in sentence s_i, and $ins(w_j)$ and $del(w_j)$ are the number of insertions and deletions of the word w_j identified in sentence s_i by the edit distance. Equation 2 defines the conditions to select a set of sentence pairs that does not contain word order errors.

$$NotWO(S) = \{s_i | w_j \in s_i; ins(w_j) \oplus del(w_j)\} \tag{2}$$

3 Data Collection

In this section we will analyze the raw data from the Lang-8 Learner Corpora[8] to extract word order errors by the edit distance method described in the previous section. The corpus contains both the original sentences written by learners and sentences corrected by other users of Lang-8 that are proficient in the target language. The learners' L1 and L2 are tagged for each document made up of a number of sentences. Firstly, we extracted sentences from the corpus that were written by Japanese L1 learners learning English that had been corrected one or more times. After removing comments and styling tags from the corrections, we then filtered to remove invalid corrections containing multiple languages which resulted in 871,432 original/corrected sentence pairs. The edit distance between the original and corrected sentence was then calculated for each of the sentence pairs. This was then analyzed to extract sentence pairs that contain word order errors, sentences that do not contain word order errors, and sentence pairs that contain a combination of errors, and therefore do not fall into either of the defined sets. The size of the extracted sets is shown in Table 1.

Error Type	# Sentences Pairs
Word order error only	7043
Other error only	742064
Word order and other error	122325

Table 1: Number of corrected sentence samples extracted.

We created a corpus for machine learning by selecting all of the sentences in the word order error set as the positive class, and then selected at random using the GNU shuf utility[1] an equal amount of sentences (7043 sentence pairs) from the other error only set as the negative class. All of the original and corrected sentences were then processed using TreeTagger[9] for Parts of Speech (POS) tagging. Words in the corrected sentence that were identified by the edit distance analysis to be either an insertion or deletion were included as both untagged and tagged words as follows: insertions were prefix tagged with "i:", deletions with"d:", and all edited words were prefix tagged with"e:". N-grams of 2 to 4 words/POS tags in length were also used for analysis. This corpus contains features from both the original and corrected sentences and we will refer to it as the *Parallel corpus*. An additional corpus containing features only from the original learner written sentences, that we will refer to as *Single corpus*, was created for the prediction of word order errors in non-corrected learner writing.

4 Word Order Error Prediction by SVM and Feature Selection

The *Parallel* and *Single* corpora were indexed using GETAssoc[2] to create a search engine for the retrieval of features and vectorization of sentence data.

4.1 Method

The SVM^{light}[10] linear kernel classifier was used for model training and evaluation. Initially an SVM model was trained on all of the corpus data only for the purpose of feature scoring. The feature score was extracted by analyzing the weights of features in the SVM model trained on all the data. The corpora were then split into train and test sets at a ratio of 9:1 for evaluation by 10-fold cross validation. The prediction performance of an SVM model trained on all of the features was evaluated as a baseline. Feature selection was then performed by selecting increasingly larger sets of N top positive and N top negative score features and evaluating the prediction performance of each set. The set with the best prediction performance is therefore the optimal feature selection.

4.2 Baseline Prediction Performance Evaluation

An SVM model trained on all features was evaluated as the baseline of prediction performance. The baseline prediction performance results are shown in Table 2 for SVM models trained by analyzing all of the features in sub-feature set of the *Parallel* corpus. The best performing SVM model by Accuracy and F-measure was trained and tested on word and POS tag features of the *Parallel* corpus. The prediction performance is high, however this is to be expected as the corpus contains features from both the original and corrected sentences along with tags indicating edits in the corrected sentence. The baseline prediction performance results for SVM models trained and tested on features from the *Single* corpus are worse as only the original learner writing

Features	F	Accuracy
Word	0.8745	0.8777
Word, N-gram	0.6184	0.7178
Word, POS	**0.9037**	**0.9043**
Word, N-gram, POS	0.3305	0.5981

Table 2: *Parallel* corpus baseline prediction performance.

[1] http://www.gnu.org/software/coreutils/shuf

[2] http://getassoc.cs.nii.ac.jp/

features are analyzed, and lacks any information on corrections made. The baseline prediction performance results are shown in Table 3, with word, N-gram, and POS tags producing the best prediction performance.

Features	F	Accuracy
Word	0.6750	0.5979
Word, N-gram	0.6813	0.5997
Word, POS	0.6839	0.6074
Word, N-gram, POS	**0.6942**	**0.6207**

Table 3: *Single* corpus baseline prediction performance.

4.3 The Effect of Feature Selection on Prediction Performance

In this section, we will examine the effectiveness of feature selection on the prediction performance of SVM models on different sub-feature sets of the corpora.

The evaluation of the optimal feature selection prediction performance on the *Parallel* corpus is shown in Table 4. Interestingly the top performing sub-feature set was that made up of words. The optimal N shows that a feature set of 1000 top positive and negative word features produces optimal prediction performance. Feature selection did not have much of an effect on the best performing baseline feature set of words and POS tags.

Features	N	F	Accuracy
Word	1000	**0.9234**	**0.9250**
Word, N-gram	40000	0.8905	0.8950
Word, POS	20000	0.9049	0.9056
Word, N-gram, POS	100000	0.8553	0.8705

Table 4: Optimal feature selection prediction performance for the *Parallel* corpus.

Features	N	F	Accuracy
Word	800	0.7115	0.6414
Word, N-gram	4000	0.7494	0.7107
Word, POS	700	0.7116	0.6625
Word, N-gram, POS	8000	**0.7509**	**0.7154**

Table 5: Optimal feature selection prediction performance for the *Single* corpus.

The optimal prediction performance for the *Single* corpus is shown in Table 5. As with the baseline prediction performance, the best prediction performance was by the sub-feature set made up of word, n-gram, and POS tag features. Optimal feature selection was achieved at an N of 8000 top positive and negative features, resulting in a gain of 0.0947 by Accuracy.

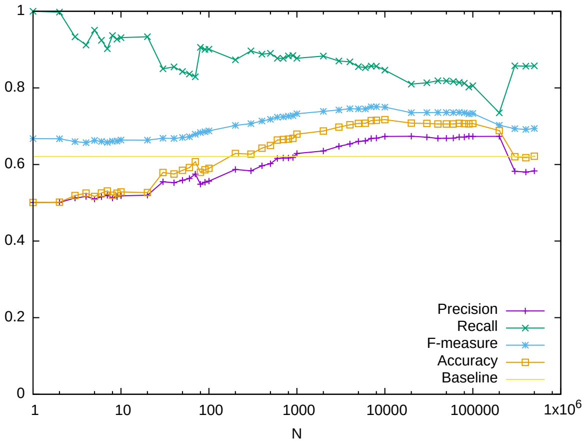

Figure 1: The effect of optimal feature selection on prediction performance for the *Single* corpus with the sub-feature set of words, n-grams, and POS tags.

A plot of the effect of feature selection relative to the baseline prediction performance is shown in Figure 1. The performance of the SVM model is greater than the baseline at $N = 200$ when measured by Accuracy. As N increases and more top positive and negative features are used for training the prediction performance by Accuracy increases until it reaches optimal feature selection at $N = 8000$. After this point, overfitting reduces the prediction performance of the SVM model as more features are added for training.

Score	Feature	Score	Feature
0.5827	rb	-0.2664	2:jj_nn
0.3514	only	-0.2130	rb_so
0.2950	2:jj_pp$	-0.2129	2:vb_dt
0.2753	2:nn_rb	-0.1950	rb_very
0.2726	more	-0.1689	very
0.2682	3:jj_pp$_nn	-0.1558	2:vbp_nn
0.2668	wrb	-0.1503	3:nn_vbz_nn
0.2666	2:jj_dt	-0.1485	never
0.2546	in_up	-0.1485	rb_never
0.2525	2:i_and	-0.1390	any

Table 6: Single corpus top 10 positive and negative features.

The top 10 positive and negative scoring features form the SVM model for the *Single* corpus is shown in Table 6. Positive scoring features are indicative of word order errors. The top scoring feature "rb" is the adverb POS tag. The POS tag bi-gram "2:jj_pp$" and "2:nn_rb" indicate combinations of adjective with possessive pronoun, and noun(singular or mass) with adverb respectively.

5 Conclusion

In this paper, we examined the use of edit distance analysis in the automatic extraction and prediction of word order errors from a Language Learning SNS. We extracted 7043 word order corrected learner writing sentence pairs from a raw corpus and combined it with 7043 randomly selected sentence pairs that do not contain word order errors to create a balanced word order error corpus for machine learning.

We then evaluated the prediction performance of an SVM model and feature selection in classifying word order errors on a *Single* and *Parallel* corpus. As expected, the results were high for the *Parallel* corpus as it contains information from the corrected sentence. The prediction performance on the *Single* corpus was improved by optimal feature selection, but further investigation is required for greater improvement. Also an evaluation of the effectiveness of extracting word order errors with the proposed method should be undertaken in future work.

6 Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 15J04830.

References

[1] Tetreault, J., Leacock, C., Automated Grammatical Error Correction for Language Learners, COLING 2014, pp. 8, 2014.

[2] Odlin, T., Cross-Linguistic Influence, The Handbook of Second Language Acquisition, pp. 436-486, 2003.

[3] Flanagan, B., Yin, C., Hashimoto, K., Hirokawa, S., Clustering English Writing Errors based on Error Category Prediction, In Proceedings of the 3rd ISEEE, pp. 733-739, 2013.

[4] Flanagan, B., Yin, C., Suzuki, T., Hirokawa, S., Intelligent Computer Classification of English Writing Errors, Intelligent Interactive Multimedia Systems and Services, Vol. 254, pp. 174-183, 2013.

[5] Tono, Y., Mochizuki, H., Toward automatic error identification in learner corpora: A DP matching approach, in Corpus Linguistics, 2009.

[6] Tono, Y., Automatic extraction of L2 criterial lexico-grammatical features across pseudo-longitudinal learner corpora: using edit distance and variability-based neighbour clustering, L2 vocabulary acquisition, knowledge and use, pp. 149-176, 2013.

[7] Levenshtein, V., Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, Vol. 10, No.8, pp. 707-710, 1966.

[8] Mizumoto, T., Komachi, M., Nagata, M., Matsumoto, Y., Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners, In IJCNLP, pp. 147-155, 2011.

[9] Schmid, H., Probabilistic part-of-speech tagging using decision trees, In Proceedings of the international conference on new methods in language processing, Vol. 12, pp. 44-49, 1994.

[10] Joachims, T., Learning to classify text using support vector machines: Methods, theory and algorithms, Kluwer Academic Publishers, 2002.