Discriminative Reranking for Grammatical Error Correction with Statistical Machine Translation

Tomoya Mizumoto
Tohoku University
tomoya-m@ecei.tohoku.ac.jp

Yuji Matsumoto
Nara Institute of Science and Technology
matsu@is.naist.jp

Abstract

Research on grammatical error correction has received considerable attention. For dealing with all types of errors, grammatical error correction methods that employ statistical machine translation (SMT) have been proposed in recent years. An SMT system generates candidates with scores for all candidates and selects the sentence with the highest score as the correction result. However, the 1-best result of an SMT system is not always the best result. Thus, we propose a reranking approach for grammatical error correction. The reranking approach is used to re-score N-best results of the SMT and reorder the results. Our experiments show that our reranking system using parts of speech and syntactic features improves performance and achieves state-of-the-art quality, with an $F_{0.5}$ score of 40.0.

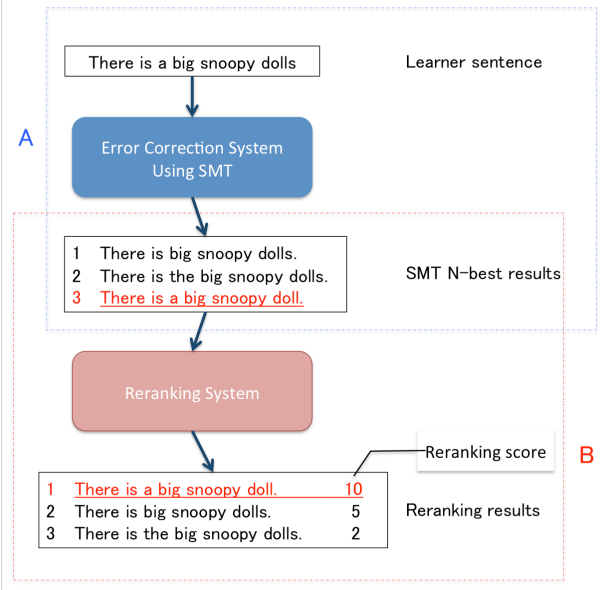

Figure 1: Flow of reranking.

1 Introduction

Research on assisting second language learners has received considerable attention, especially regarding grammatical error correction of essays written by English as a Second Language (ESL) learners. To address all types of errors, grammatical error correction methods that use statistical machine translation (SMT) have been proposed (Brockett et al., 2006; Mizumoto et al., 2012; Buys and van der Merwe, 2013; Yuan and Felice, 2013; Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014). SMT-based error correction systems have achieved rankings first and third in the CoNLL2014 Shared Task (Ng et al., 2014).

SMT systems generate many candidates of translation. SMT systems generate scored candidates and select a sentence having the highest score as the translation result. However, the 1-best result of SMT system is not always the best result because the scoring is conducted only with local features. In other words, N-best (N $>$ 1) results may be better than the 1-best result.

Reranking approaches have been devised to solve the scoring problem. Reranking is a method that re-scores N-best candidates of SMT and reorders the candidates by score. Figure 1 shows a flow of reranking. First, N-best results are obtained by a grammatical error correction system using SMT for a learner sentence (A in Figure 1). A reranking sys-

Proceedings of NAACL-HLT 2016, pages 1133–1138,
San Diego, California, June 12-17, 2016.

tem then re-scores the N-best results and reorders them (B in Figure 1).

In this study, we apply a discriminative reranking method to the task of grammatical error correction. Syntactic information is not considered in the phrase-based SMT. We show that using syntactic features in the reranking system can improve error correction performance. Although reranking using only surface features (Shen et al., 2004) is not effective for grammatical error correction, reranking using syntactic features improves the $F_{0.5}$ score.

2 Related Work for Reranking

Reranking approaches have been proposed for common SMT tasks (Shen et al., 2004; Carter and Monz, 2011; Li and Khudanpur, 2008; Och et al., 2004). Shen et al. (2004) first used a perceptron-like algorithm for reranking of common SMT tasks. However, they used only a few features.

Li and Khudanpur (2008) proposed a reranking approach that uses a large-scale discriminative N-gram language model for common SMT tasks. They extended the reranking method for automatic speech recognition (Roark et al., 2007) to SMT tasks. The approach of Carter and Monz (2011) was similar to that of Li and Khudanpur (2008), but they used additional syntactic features (e.g. part of speech (POS), parse tree) for reranking of common SMT tasks.

The reranking approach has been used in grammatical error correction based on phrase-based SMT (Felice et al., 2014). However, their method uses only language model scores. In the reranking step, the system can consider not only surface but also syntactic features such as those in the approach of Carter and Monz (2011). We use syntactic features in our reranking system.

Heafield et al. (2009) proposed a system combination method for machine translation that is similar to that of reranking. System combination is a method that merges the outputs of multiple systems to produce an output that is better than each individual system. Susanto et al. (2014) applied this system combination to grammatical error correction. They combined pipeline systems based on classification approaches and SMT systems. Classifier-based systems use syntactic features as POS and dependency for error correction. However, syntactic information is not considered in combining systems.

Table 1: Oracle score of grammatical error correction

N-best	Precision	Recall	$F_{0.5}$
1	43.9	24.5	37.9
10	79.1	36.7	64.3
50	89.5	43.1	73.6
100	92.3	45.3	76.4

3 Why is Reranking Necessary?

Grammatical error correction using SMT has the same problem as that of common SMT task: the 1-best correction by the system is not always the best. To prove this, we conducted a grammatical error correction experiment using SMT and calculated N-best oracle scores. The oracle scores are calculated by selecting the correction candidate with the highest score from the N-best results for each sentence.

Table 1 shows oracle scores of a baseline grammatical error correction system using SMT[1]. Althogh the $F_{0.5}$ score of the 1-best output was 37.9, the $F_{0.5}$ of the 10-best oracle score was 64.3. The higher the value of N-best, the higher is the oracle score. This result reveals that the 1-best correction by a grammatical error correction system using SMT is not always the best.

Advantage of Reranking Two advantages exist for using a reranking approach for grammatical error correction. The first is that a reranking system can use POS and syntactic features unlike phrase-based SMT. With some errors, the relation between distant words must be considered (e.g., article relation between *a* and *dolls* in the phrase *a big Snoopy dolls)*.

The second advantage is that POS taggers and parsers can analyze error-corrected candidates more properly than they analyze erroneous sentences, which enables more accurate features to be obtained. Thus, the fact that taggers for N-best corrected results work much better than for learner original sentences is promising.

4 Proposed Method

In this section, we explain our discriminative reranking method and features of reranking for grammati-

[1] See 5.1 for a baseline system

Table 2: Features for reranking. Examples show features for the sentence *I agree with this statement to a large extent*. The features excluding "Web dependency N-gram" are binary valued. "Web dependency N-gram" is unit interval [0,1] valued.

Feature name	Examples
Word 2,3-gram	I agree; I agree with; agree with; agree with this; this statement
POS 2,3,4,5-gram	PRP VBP; PRP VBP IN; PRP VBP IN DT; PRP VBP IN DT NN
POS-function word 2,3,4,5-gram	PRP VBP; PRP VBP with; PRP VBP with this; PRP VBP with this NN
Web dependency N-gram	prep-agree-with-statment; det-a-extent

cal error correction.

4.1 Discriminative Reranking Method

In this study, we use a discriminative reranking algorithm using perceptron which successfully exploits syntactic features for N-best reranking for common translation tasks (Carter and Monz, 2011). Figure 2 shows the standard perceptron algorithm for reranking. In this figure, T is the number of iterations for perceptron learning and N is the number of learner original sentences in the training corpus. In addition, $GEN(x)$ is the N-best list generated by a grammatical error correction system using SMT for an input sentence and $ORACLE(x^i)$ determines the best correction for each of the N-best lists according to the $F_{0.5}$ score. Moreover w is the weight vector for features and ϕ is the feature vector for candidate sentences. When selecting the sentence with the highest score from candidate sentences (line 5), if the selected sentence matches oracle sentence, then the algorithm proceeds to next sentence. Otherwise, the weight vector is updated.

The disadvantage of perceptron is instability when training data are not linearly separable. As a solution to this problem, an averaged perceptron algorithm was proposed (Freund and Schapire, 1999). In this algorithm, weight vector w_{avg} is defined as:

$$w_{avg} = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N}\sum_{i=1}^{N} w_t^i \quad (1)$$

To select the best correction from N-best candidates, we use the following formula:

$$S(z) = \beta\phi_0(z) + w\cdot\phi(z) \quad (2)$$

where $\phi_0(z)$ is the score calculated by the SMT system for each translation hypothesis. This score is weighted by β. Using $\phi_0(z)$ as a feature in the perceptron algorithm is possible, but this may lead to

1: $w \leftarrow 0$
2: **for** $t = 1$ to T **do**
3: **for** $i = 1$ to N **do**
4: $y^i \leftarrow ORACLE(x^i)$
5: $z^i \leftarrow argmax_{x\in GEN(x^i)}\phi(z)\cdot w$
6: **if** $z^i \neq y^i$ **then**
7: $w \leftarrow w + \phi(y^i) - \phi(z^i)$
8: **end if**
9: **end for**
10: **end for**
11: **return** w

Figure 2: Perceptron algorithm for ranking.

under-training (Sutton et al., 2006). We select the value for β with the highest $F_{0.5}$ score by changing β from 0 to 100 in 0.1 increments on the development data.

4.2 Features of Discriminative Reranking for Grammatical Error Correction

In this study, we use the features used in Carter and Monz (2011) as well as our new features of POS and dependency. We use the features extracted from the following sequences: POS tag , shallow parse tag, and shallow parse tag plus POS tag sequences (Carter and Monz, 2011). From these sequences, features are extracted based on the following three definitions:

1. $(t_{i-2}t_{i-1}t_i)$, $(t_{i-1}t_i)$, (t_iw_i)
2. $(t_{i-2}t_{i-1}w_i)$
3. $(t_{i-2}w_{i-2}t_{i-1}w_{i-1}t_iw_i)$, $(t_{i-2}t_{i-1}w_{i-1}t_iw_i)$, $(t_{t-1}w_{i-1}t_iw_i)$, $(t_{i-1}t_iw_i)$

Here, w_i is a word at position i and t_i is a tag (POS or shallow parse tag) at position i.

Table 2 shows our new features. For the "POS-function N-gram" feature, if words are con-

Table 3: Experimental results. TP, FN, and FP denote true positive, false negative, and false positive, respectively.Asterisks indicate that the difference between the baseline and reranking results is statistically significant ($p < 0.01$, bootstrap test).

		Precision	Recall	$F_{0.5}$	TP	FN	FP	GLEU
Baseline								
1	1-best result of SMT	43.9	24.5	37.9	598	1847	764	65.7
2	Reranking by N-gram LM	39.5	**31.7**	37.6	834	1797	1280	64.7
3	CAMB (CoNLL2014)	39.7	30.1	37.3	772	1793	1172	64.5
4	CUUI (CoNLL2014)	41.8	24.9	36.8	623	1881	868	64.8
Discriminative reranking								
5	Word 2,3-gram	43.7	24.8	37.9	606	1834	781	65.7
6	Features of Carter (2011)	44.3	26.7	39.1	669	1837	842	65.8
7	Our features (Table 2)	**45.8**	26.6	**40.0***	657	1813	778	**66.1**
8	All features (6+7)	44.4	27.1	39.4*	679	1827	851	65.8

tained in a stop word list, we use surface form, otherwise we use POS tags. "Web dependency N-gram" is feature used in Dahlmeier et al. (2012). We collect log frequency counts for dependency N-grams from a large dependency-parsed web corpus and normalize all real-valued feature values to a unit interval [0,1].

5 Experiments of Reranking

We conducted experiments on grammatical error correction to observe the effect of discriminative reranking and our syntactic features.

5.1 Experimental Settings

We used phrase-based SMT which many previous studies used for grammatical error correction for a baseline system. We used cicada 0.3.5[2] for the machine translation tool and KenLM[3] as the language modeling tool. We used ZMERT[4] as the parameter tuning tool and implemented the averaged perceptron for reranking.

The translation model was trained on the Lang-8 Learner Corpora v2.0. We extracted English essays that were written by ESL learners and cleaned noise with the method proposed in (Mizumoto et al., 2011). From the results, we obtained 1,069,127 sentence pairs. We used a 5-gram language model built on the "Associated Press Worldstream English Service" from English Gigaword corpus and NUCLE 3.2 (Dahlmeier et al., 2013). We used these two language models as separate feature functions in the SMT system.

For training data of reranking, Lang-8 Learner Corpora was split into 10 parts and each part was corrected by a grammatical error correction system trained on the other nine parts. We selected 10 as N for N-best reranking. PukWaC corpus (Baroni et al., 2009) was used for constructing our "Web dependency N-gram" feature. We use Stanford Parser 3.2.0[5] as a dependency parser.

CoNLL-2013 test set were split into 700 sentences for parameter tuning of SMT and 681 sentences for tuning parameter beta. CoNLL-2014 test set, 1,312 sentences were used for evaluation. We used M2 Scorer as an evaluation tool (Dahlmeier and Ng, 2012). This scorer calculates precision, recall, and $F_{0.5}$ scores. We used $F_{0.5}$ as a tuning metric. In addition, we used GLEU (Napoles et al., 2015) as evaluation metrics.

5.2 Experimental Results and Discussion

Table 3 shows the experimental results. We used the 1-best result of the SMT correction system and reranking by probability of the large N-gram language model (Felice et al., 2014) as baseline systems. In addition, we compared the systems that are ranked first (CAMB) and second (CUUI) (Felice et al., 2014; Rozovskaya et al., 2014) in CoNLL2014

[2] http://www2.nict.go.jp/univ-com/multi_trans/cicada/

[3] https://kheafield.com/code/kenlm/

[4] http://cs.jhu.edu/~ozaidan/zmert/

[5] http://nlp.stanford.edu/software/lex-parser.shtml

Shared Task.

The discriminative reranking system with our features achieved the best $F_{0.5}$ score. The difference between the results of baseline and reranking using our features was statistically significant ($p < 0.01$). Because a large N-gram language model was adopted for reranking, recall increased considerably but precision declined. This result is extremely similar to that of the CAMB system, which is an SMT-based error correction system that reranks by using a large N-gram language model. When we compare the reranking system using our features to CUUI, our system is better in all metrics.

When we use the discriminative reranking with our features, both precision and recall increase. In the experimental results of system combination (Susanto et al., 2014), recall increases but precision declines with respect to original SMT results. In addition, precision increases but recall declines with respect to pipeline results.

The reranking that employed all features generated a lower $F_{0.5}$ score than when only our features were used. One reason for this is that the roles of features overlap. These experiments revealed that reranking is effective in grammatical error correction tasks and that POS and syntactic features are important.

6 Conclusion

We proposed a reranking approach to grammatical error correction using phrase-based SMT. Our system achieved $F_{0.5}$ score of 40.0 (an increase of 2.1 points from that of the baseline system) on the CoNLL2014 Shared Task test set. We showed that POS and dependency features are effective for the reranking of grammatical error correction.

In future work, we will use the adaptive regularization of weight vectors (AROW) algorithm (Crammer et al., 2009) instead of the averaged perceptron. In addition, we will apply the pairwise approach to ranking (Herbrich et al., 1999) used in information retrieval to rerank of grammatical error correction.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

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

Jan Buys and Brink van der Merwe. 2013. A Tree Transducer Model for Grammatical Error Correction. In *Proceedings of CoNLL Shared Task*, pages 43–51.

Simon Carter and Christof Monz. 2011. Syntactic Discriminative Language Model Rerankers for Statistical Machine Translation. *Machine Translation*, 25(4):317–339.

Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive Regularization of Weight Vectors. In *NIPS*, pages 414–422.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better Evaluation for Grammatical Error Correction. In *Proceedings of NAACL-HLT*, pages 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. 2012. NUS at the HOO 2012 Shared Task. In *Proceedings of BEA*, pages 216–224.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of BEA*, pages 22–31.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of CoNLL Shared Task*, pages 15–24.

Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296.

Kenneth Heafield, Greg Hanneman, and Alon Lavie. 2009. Machine translation system combination with flexible word ordering. In *Proceedings of Workshop on SMT*, pages 56–60.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. 1999. Support Vector Learning for Ordinal Regression. In *Proceedings of ICANN*, pages 97–102.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. The AMU System in the CoNLL-2014 Shared Task: Grammatical Error Correction by Data-Intensive and Feature-Rich Statistical Machine Translation. In *Proceedings of CoNLL Shared Task*, pages 25–33.

Zhifei Li and Sanjeev Khudanpur. 2008. Large-scale Discriminative n-gram Language Models for Statistical Machine Translation. In *Proceedings of AMTA*.

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.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2012. The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings. In *Proceedings of COLING*, pages 863–872.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground Truth for Grammatical Error Correction Metrics. In *Proceedings of ACL-IJCNLP*, pages 588–593.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of CoNLL Shared Task*, pages 1–14.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of HLT-NAACL*, pages 161–168.

Brian Roark, Murat Saraclar, and Michael Collins. 2007. Discriminative n-gram language modeling. *Computer Speech Language*, 21(2):373–392.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia System in the CoNLL-2014 Shared Task. In *Proceedings of CoNLL Shared Task*, pages 34–42.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative Reranking for Machine Translation. In *Proceedings of HLT-NAACL*.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of EMNLP*, pages 951–962.

Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing Weight Undertraining in Structured Discriminative Learning. In *Proceedings of HLT-NAACL*, pages 89–95.

Zheng Yuan and Mariano Felice. 2013. Constrained Grammatical Error Correction using Statistical Machine Translation. In *Proceedings of CoNLL Shared Task*, pages 52–61.