Exploiting N-Best Hypotheses to Improve an SMT Approach to Grammatical Error Correction

Duc Tam Hoang,[1] **Shamil Chollampatt,**[2] and **Hwee Tou Ng**[1,2]

[1]Department of Computer Science

[2]NUS Graduate School for Integrative Sciences and Engineering

National University of Singapore

{hoangdt, nght}@comp.nus.edu.sg, shamil@u.nus.edu

Abstract

Grammatical error correction (GEC) is the task of detecting and correcting grammatical errors in texts written by second language learners. The statistical machine translation (SMT) approach to GEC, in which sentences written by second language learners are translated to grammatically correct sentences, has achieved state-of-the-art accuracy. However, the SMT approach is unable to utilize global context. In this paper, we propose a novel approach to improve the accuracy of GEC, by exploiting the n-best hypotheses generated by an SMT approach. Specifically, we build a classifier to score the edits in the n-best hypotheses. The classifier can be used to select appropriate edits or re-rank the n-best hypotheses. We apply these methods to a state-of-the-art GEC system that uses the SMT approach. Our experiments show that our methods achieve statistically significant improvements in accuracy over the best published results on a benchmark test dataset on GEC.

1 Introduction

Using statistical machine translation (SMT) for the task of grammatical error correction (GEC) has gained popularity as one of the most promising approaches. In general, language learners can make different types of grammatical errors. It becomes difficult to target each error type with traditional classification-based approaches that use hand-crafted features for correcting specific error types. On the other hand, SMT based systems have the ability to correct a wide variety of error types without language-specific features. In order to use the SMT framework, the task of GEC is formulated as a translation task from *bad* English to *good* English. The SMT system, which is trained using parallel corpora (the erroneous sentences and their corresponding corrected sentences), attempts to output the corrected target sentence given an erroneous source sentence.

Phrase-based SMT components have been used in state-of-the-art grammatical error correction systems [Susanto *et al.*, 2014]. It has been shown that increasing the amount of parallel data further improves the performance of SMT-based GEC systems [Junczys-Dowmunt and Grundkiewicz, 2014].

However, SMT systems excessively rely on parallel data and tend to correct errors depending on the frequencies of corrections observed in the training data, without considering a global context. The error correction can be further improved if it had access to a larger context spanning phrase boundaries. Specifically, the phrase-based SMT approach does not capture long-distance dependency between words that are further apart in a sentence. As a consequence, the system will make several independent corrections, yet produce an overall ungrammatical sentence.

Table 1 shows a few examples of invalid corrections made by the SMT approach. In the first example, *friends feels lonely* is corrected to *friends feel lonely*. Although this correction seems accurate locally, it is actually invalid considering the global context in the sentence. Specifically, the SMT approach has limitations in correcting grammatical errors like subject-verb agreement, article-noun, and verb-form errors. The examples in Table 1 show that taking the global context into consideration can improve the SMT approach to GEC.

In this paper, we improve the correction made by an SMT approach to GEC, by incorporating a larger context and adding linguistic information. Our *novel* approach exploits the top n correction candidates (*n-best* hypotheses) generated by the SMT system. We build an *edit classifier* to validate the edits made by the SMT system. The features of this classifier use the available global context from a sentence. We propose two different ways of using this classifier to improve the corrections made by the SMT system:

1. ***Reranking***: We rerank the n-best hypotheses by assigning a score to each hypothesis. The score is used as a feature along with other SMT features in order to rerank the n-best hypotheses.

2. ***Edit selection***: All the edits in the n-best hypotheses are scored by the classifier. We then select the highest scoring edits from different hypotheses and apply them on the source sentence in order to generate a better hypothesis.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes the *edit classifier*. Section 4 describes the two methods to exploit the n-best hypotheses. Section 5 presents the experimental setup and results. Section 6 discusses the results. Finally, Section 7 gives the conclusion.

#	Description	Sentence
1	*input*	The man with a huge number of friends feels lonely .
	reference	The man with a huge number of friends feels lonely .
	hypothesis	The man with a huge number of friends <u>feel</u> lonely .
2	*input*	He carries a gun into his pocket and walk into the bar .
	reference	He carries a gun <u>in</u> his pocket and <u>walks</u> into the bar .
	hypothesis	He carries a gun <u>in</u> his pocket and <u>walking</u> into the bar .
3	*input*	He has an carved green Venetian glass salad bowls .
	reference	He has ___ carved green Venetian glass salad bowls .
	hypothesis	He has <u>a</u> carved green Venetian glass salad bowls .
4	*input*	The train crashed and all passengers were died .
	reference	The train crashed and all passengers ___ died .
	hypothesis	The train <u>crashes</u> and all passengers ___ died .

Table 1: Examples of errors made by an SMT approach to GEC.

2 Related Work

Earlier approaches to GEC aimed at building classifiers or rule-based systems targeting specific error types. More recently, the SMT approach to GEC has gained attention. Brockett *et al.* [2006] used an SMT approach to correct errors related to countability of mass nouns, which poses problems to English as Second Language (ESL) learners. Mizumoto *et al.* [2011] used an SMT approach that focuses on correcting grammatical errors of Japanese learners. They used large-scale data mined from the language learning website, Lang-8[1] in order to train their SMT system. They later implemented an SMT system for GEC in English [Tajiri *et al.*, 2012]. The availability of large-scale error annotated data for GEC further increased the popularity of the SMT approach to GEC. Junczys-Dowmunt and Grundkiewicz [2014] performed various experiments using an SMT approach to GEC, such as increasing the size of the parallel corpora, using web-scale language models, and varying the methods for tuning the SMT system.

Several shared tasks have been organized in recent years for the GEC task. The CoNLL-2013 shared task [Ng *et al.*, 2013] focuses on correction of five error types: determiner, preposition, noun number, verb form, and subject-verb agreement errors. Yuan and Felice [2013] added other learner corpora to the provided training data. They used a phrase-based SMT system for correction of all five error types. Meanwhile, Yoshimoto *et al.* [2013] used an SMT approach for only two error types, prepositions and determiners. For other error types, they reported that the classification approach performs better. The CoNLL-2014 shared task [Ng *et al.*, 2014] focused on correction of all error types. The number of teams using the SMT approach increased. They achieved competitive results compared to other teams which used the classification approach [Rozovskaya *et al.*, 2014].

Recent work by Susanto *et al.* [2014] combined the outputs of multiple systems based on both the SMT approach and the classification approach. They used MEMT [Heafield and Lavie, 2010] to combine four systems (two classification-based systems and two SMT-based systems) to achieve an

$F_{0.5}$ score of 39.39. Yuan and Briscoe [2016] later proposed a neural machine translation approach for grammatical error correction. They achieved a higher $F_{0.5}$ score of 39.90 on the CoNLL-2014 test set, but using a much larger training set.

More recently, discriminative reranking methods for SMT-based GEC have been proposed [Mizumoto and Matsumoto, 2016; Yuan *et al.*, 2016]. Yuan *et al.* [2016] use a ranking SVM to rerank the n-best list without employing any syntactic features. Syntactic features and long-distance dependency relationships are important cues to assess the grammaticality of a sentence. Hence, we make use of syntactic features like constituency parse and part-of-speech tags in our approach. Mizumoto and Matsumoto [2016] also perform discriminative reranking with a perceptron using syntactic features. However, they only include the final score given by the SMT system in reranking and do not include all the other SMT features. On the other hand, we perform reranking using all SMT features and an additional feature given by our edit classifier. Most importantly, both Mizumoto and Matsumoto [2016] and Yuan *et al.* [2016] focus solely on reranking the n-best hypotheses and do not generate any new hypothesis, unlike our *edit selection* method.

3 Edit Classifier

In general, a grammatical error correction system takes an input *source* sentence and produces a corrected *hypothesis* sentence. We extract the corrections made by the system, which we refer to as *edits*. For example:
Source:
He carries a gun <u>into</u> his pocket and <u>walk</u> into the bar.
Hypothesis:
He carries a gun <u>in</u> his pocket and <u>walking</u> into the bar.

The *edits* in the above example are $(into \rightarrow in)$ and $(walk \rightarrow walking)$. We build a binary classifier to classify an *edit* as valid or invalid. A valid edit is a good correction and an invalid edit is a bad correction made by the system. In this example, the first edit is valid while the second is not. Section 3.1 describes the features used to train the classifier and Section 3.2 describes the training algorithm.

3.1 Features

We select features to build a binary classifier to distinguish between valid and invalid edits. We use both categorical and numerical features. If the same edit occurs in more than one hypothesis, the features are extracted from the highest ranked hypothesis in the n-best list. Table 2 shows the detailed features using an illustrating example. We use a 5-gram language model trained on English Wikipedia (approximately 1.78 billion words). We use the Stanford parser [Klein and Manning, 2003] for constituency parsing.

Features	Example
SMT Features	
rank of the hypothesis	1
Lexical and POS Features	
source phrase	$waits$
hypothesis phrase	sat
source + hyp. phrase	$waits{+}sat$
POS of source phrase	VBZ
POS of hyp. phrase	VBD
POS of source + hyp. phrase	$VBZ{+}VBD$
Context Features	
word before + source phrase	$cat{+}waits$
word before + hyp. phrase	$cat{+}sat$
source phrase + word after	$waits{+}on$
hyp. phrase + word after	$sat{+}on$
nearest NP head to the left	cat
+ (source, hyp. phrase)	$cat{+}\{waits, sat\}$
nearest NP head to the right	dog
+ (source, hyp. phrase)	$dog{+}\{waits, sat\}$
nearest VP head to the left	$NULL$
+ (source, hyp. phrase)	$NULL{+}\{waits, sat\}$
nearest VP head to the right	$eats$
+ (source, hyp. phrase)	$eats{+}\{waits, sat\}$
Language Model Features	
source sentence	$log\,LM(src)$
hypothesis sentence	$log\,LM(hyp)$
source phrase	$log\,LM(waits)$
hypothesis phrase	$log\,LM(sat)$
word before+source phrase	$log\,LM(cat\ waits)$
word before+hyp. phrase	$log\,LM(cat\ sat)$
source phrase+word after	$log\,LM(waits\ on)$
hyp. phrase + word after	$log\,LM(sat\ on)$
score difference	$log\,LM(sat) - logLM(waits)$ $log\,LM(hyp) - logLM(src)$

Table 2: Features for the edit classifier. Example: edit $waits \rightarrow sat$ from the edits *"the cat {$\underline{waits}{\rightarrow}\underline{sat}$} on the dog and eats a mouse ."*

There are four main categories of features which are used to build the model, described as follows:

SMT Features

We use the rank of the hypothesis in the n-best list in which the edit occurs as a feature. Our intuition is that a hypothesis with a higher rank may contain more valid edits than one with a lower rank.

Lexical and POS Features

We use the words occurring in the source side and target side of an edit and their parts-of-speech (POS) as features. The lexical features can determine the choice and order of words and the POS features can determine the grammatical roles of words in the edit within a hypothesis.

Context Features

Context features correspond to how the edits interact with other words of the sentence. They capture the context information surrounding the current edit, ranging from neighboring words (preceding and following words) to constituents (heads of noun phrases and verb phrases). The neighboring words provide collocation information. We identify the interaction between the edit and constituents by the nearest noun phrase (or verb phrase) head to the left (or right) of the edit. For example: between verb and preposition (*send a huge package to my friend*), between subject and verb (*people with albinism are prone to sunburn*), and between verb forms (*He opened the windows and she felt cold*).

Language Model Features

Additionally, we include the language model scores for the source and target phrases of the *edit* and the source sentence and hypothesis sentence after applying the *edit*. The language model features will help to ensure fluency of the hypothesis sentence after applying the *edit*.

3.2 Training the Classifier

In order to obtain edits to train the classifier, we use an SMT-based GEC system to translate the source side of an ESL corpus and obtain the n-best hypotheses for each source sentence. We use the NUS Corpus of Learner English (NUCLE) [Dahlmeier *et al.*, 2013] as the ESL corpus. We obtain the edits of each hypothesis for every source sentence. The edits which match the gold edits are considered valid and the other edits are considered invalid. Gold edits are obtained by comparing a corrected sentence by human to the original source sentence written by an ESL learner.

We choose confidence-weighted learning [Dredze *et al.*, 2008] to train the *edit classifier*. Confidence-weighted (CW) learning is an online learning algorithm which outputs a real number corresponding to the confidence in the prediction. CW weighted classifiers perform well in cases where the feature space is high dimensional and sparse, making it a popular choice for natural language processing tasks which use n-grams as features. Furthermore, to improve our classifier, we tuned a threshold for our classifier using the CoNLL-2013 dataset and performed a grid search for values of threshold from –0.5 to 0.5 with a step size of 0.01.

4 Exploiting N-Best Hypotheses

We use a phrase-based SMT approach for GEC. It follows the log-linear model formulation [Och and Ney, 2002]:

$$\hat{t} = \arg\max_t \exp\Big(\sum_{i=1}^{n} \lambda_i h_i(s, t)\Big) \quad (1)$$

In Equation 1, s is the source sentence, t is a hypothesis sentence, n is the number of features, h_i is a feature function, and λ_i is its weight. The feature weights λ_i is tuned by minimum error rate training (MERT) [Och, 2003] on a development set optimizing the $F_{0.5}$ measure [Junczys-Dowmunt and Grundkiewicz, 2014]. The $F_{0.5}$ measure is a popular metric

used for GEC [Dahlmeier and Ng, 2012] and has been used in the CoNLL-2014 shared task as the official evaluation metric.

The SMT decoder finds the best hypothesis $\hat{t}$ according to Equation 1. Alternatively, the SMT system can produce a list of n-best hypotheses ranked by the log-linear model. We propose two methods of using these n-best hypotheses to produce a better corrected output: *reranking* and *selection of edits*. Reranking is described in Section 4.1 and edit selection in Section 4.2

4.1 Reranking

A potential use of the edit classifier is to rerank the SMT n-best hypotheses. We obtain the n-best hypotheses using an SMT system. We augment the list of features for each hypothesis with the average score given by the edit classifier for all edits in the hypothesis. The new feature function h_{n+1} is given by the following equation:

$$h_{n+1}(s, t) = \frac{1}{|E|} \sum_{e \in E} score(e) \qquad (2)$$

where s and t are the source sentence and the hypothesis sentence respectively, E is the set of edits associated with hypothesis t, and $score(e)$ is the score assigned to the edit e by the edit classifier. The weight λ_{n+1} for this feature function is optimized along with the other SMT features by running MERT optimizing $F_{0.5}$ on the development set. We then rescore and rerank the hypotheses with the new set of features and weights.

4.2 Edit Selection

The edit classifier is used to classify an edit as valid or invalid. We discard all invalid edits. We adopt a simple scheme to select a subset of valid edits. From the set of all valid edits of all n-best hypotheses, we first select the edit which has the highest score assigned by the edit classifier. We then select the next highest scoring edit from the set, on the condition that it does not overlap with any edits selected so far. Two edits overlap if the source phrases of the two edits have some overlapping words in the source sentence. We repeat this process until we cannot find any additional non-overlapping edit. Compared to the reranking method which is restricted to reranking the top n hypotheses but cannot generate a new hypothesis sentence, the edit selection method is capable of generating a new hypothesis sentence by selecting edits from different hypotheses.

5 Experiments

We empirically evaluate our proposed methods in the context of the CoNLL-2014 shared task. Description of the shared task can be found in the overview paper [Ng *et al.*, 2014]. In this section, we only describe the salient details related to our work.

5.1 Setup

We build a baseline system using Moses [Koehn *et al.*, 2007] with the standard configuration for phrase-based SMT. Two parallel corpora are used for training the translation model.

The first corpus is the NUS Corpus of Learner English (NUCLE) [Dahlmeier *et al.*, 2013]. The second corpus is the Lang-8 Corpus of Learner English v1.0 [Tajiri *et al.*, 2012]. We train one phrase table using the concatenation of the two corpora. Word alignments are obtained using GIZA++ [Och and Ney, 2003] and phrase pairs are extracted using the standard phrase extraction tool provided by Moses. Statistics of the training data can be found in Table 3.

We train two 5-gram language models using the KenLM language modeling toolkit [Heafield, 2011]. The first language model is trained on the target side of NUCLE. The second language model is trained on English Wikipedia (approximately 1.78 billion words). The two language models are used as two separate features in the log-linear model.

We do not use any reordering model as most error types do not involve long-distance reordering. Local reordering is generally captured by phrase pairs in the phrase table. The distortion limit is set to 0 to prevent reordering during the hypothesis generation phase. Tuning is done on the CoNLL-2013 data. We use MERT [Och, 2003] as the tuning method.

Dataset	# sentences	# source tokens	# target tokens
NUCLE	57,151	1,161,567	1,155,559
Lang-8 v1.0	1,114,139	12,945,666	13,232,058
CoNLL-2013	1,381	29,207	28,743
CoNLL-2014	1,312	30,114	29,881 30,229

Table 3: Statistics of training, development, and test data. The CoNLL-2014 test data has corrected sentences by two human annotators, hence it has two target token counts, one for the corrections made by each annotator.

Finally, the system is evaluated on the CoNLL-2014 test set. Statistics of the development and test data can be found in Table 3, in which the CoNLL-2014 test data has corrected sentences by two human annotators. We use the $F_{0.5}$ measure as the evaluation metric for both tuning and testing. $F_{0.5}$ weights precision twice as much as recall. Given a set of sentences, where G_s is the set of gold edits in the annotation for sentence s, and E_s is the set of system edits for sentence s, precision, recall and $F_{0.5}$ are defined as follows:

$$P = \frac{\sum_s |G_s \cap E_s|}{\sum_s |E_s|} \qquad (3)$$

$$R = \frac{\sum_s |G_s \cap E_s|}{\sum_s |G_s|} \qquad (4)$$

and

$$F_{0.5} = \frac{(1 + 0.5^2) \times R \times P}{R + 0.5^2 \times P} \qquad (5)$$

where $G_s \cap E_s$ is defined as:

$$G_s \cap E_s = \{e \in E_s \mid \exists g \in G_s, match(g, e)\} \qquad (6)$$

We use the M²Scorer [Dahlmeier and Ng, 2012] for evaluation, which was the official scorer for the CoNLL-2014 shared task. The scorer determines the system edits that have

maximal overlap with the gold edits. For the statistical significance test, we use one-tailed sign test with bootstrap resampling on 100 samples.

5.2 Results

We first evaluate our edit classifier separately, independent of the entire GEC system. In order to do this, we compute the accuracy of the classifier on the edits obtained from the CoNLL-2013 dataset. Since we use the CoNLL-2013 dataset for threshold tuning, we evaluate the classifier before threshold tuning. We perform ablation testing, by removing each category of features: SMT features, lexical and POS features, context features, and language model features. The results of our experiments are summarized in Table 4, showing that each group of features contributes to the performance of our classifier.

Features	Accuracy
All	**61.25**
without SMT features	61.17
without lexical and POS features	61.01
without context features	60.92
without LM features	59.93

Table 4: Feature ablation test.

Table 5 shows the performance of the baseline SMT system on the CoNLL-2014 test set. An important point is that the baseline system outperforms all state-of-the-art systems on the CoNLL-2014 test set, making it a highly competitive baseline. Note that our baseline system uses exactly the same training data as [Susanto *et al.*, 2014] for training the translation model and the language model. The difference between our baseline system and the SMT components of [Susanto *et al.*, 2014] is that we tune with $F_{0.5}$ instead of BLEU and we use the standard Moses configuration without the Levenshtein distance feature.

System	P	R	$F_{0.5}$
Baseline	50.56	22.68	**40.58**
State-of-the-art systems			
Mizumoto and Matsumoto [2016]	45.80	26.60	40.00
Yuan and Briscoe [2016]	-	-	39.90
Susanto *et al.* [2014]	53.55	19.14	39.39
Yuan *et al.* [2016]	-	-	38.08
Top 3 systems in the CoNLL-2014 shared task			
CAMB	39.71	30.10	37.33
CUUI	41.78	24.88	36.79
AMU	41.62	21.40	35.01

Table 5: Performance of the baseline system and other state-of-the-art systems on the CoNLL-2014 test set.

We perform reranking and edit selection using the edit classifier. The experimental results on the n-best hypotheses generated by the SMT baseline on the CoNLL-2014 test set are shown in Table 6. The reranking method is applied on the 5-best and 10-best hypotheses. The edit selection method selects non-overlapping edits from the top n hypotheses with $n \in [1, 5]$.

System	P	R	$F_{0.5}$
Baseline	50.56	22.68	40.58
Reranking			
5-best	50.32	22.99	40.65
10-best	50.79	22.92	**40.85**[*]
Edit selection			
1-best	51.22	22.28	40.66
2-best	50.35	23.70	41.11
3-best	50.31	23.82	41.16
4-best	50.31	23.82	41.16
5-best	50.35	23.84	**41.19**[*]

Table 6: Performance of reranking and edit selection on the CoNLL-2014 test set. [*] indicates that the improvements are statistically significant ($p < 0.01$) compared to the baseline system.

We observe that both the reranking method and the edit selection method are able to obtain improvements compared to the baseline. It is worth noting that the reranking method only rescores the n-best hypotheses and selects the hypothesis which attains the highest score. The edit selection method can be seen as a combination of multiple hypotheses, since it is capable of producing a new hypothesis different from any of the top n hypotheses. Our experimental results show that the edit selection method performs better than the reranking method.

Our experiments also show that there is no further improvement in reranking top k hypotheses for $k > 10$. This is because most of the source sentences produce less than 10 hypotheses. This happens because Moses prunes away a hypothesis during decoding if it gets a low score by the log-linear model. In addition, our analysis shows that lower-ranked hypotheses often contain edits which receive low scores by the edit classifier. Since the average classifier score is used as an additional feature during reranking, lower-ranked hypotheses do not help to improve performance.

6 Discussion

In this section, we discuss the strengths and weaknesses of the SMT approach, showing how our edit selection method helps to improve the system performance. Table 7 shows a few examples which compare the output of the edit selection method to the output of the baseline system.

Both the subject and verb must agree in number in a sentence. The phrase-based SMT approach to GEC does not take into consideration the interaction between the subject and verb when they are located far from each other. This makes it difficult for the SMT-based baseline to correct such errors. The baseline system also introduces subject-verb agreement errors while attempting to make corrections. In contrast, our edit classifier considers the interaction between an edit and other phrases and can utilize the n-best hypotheses to produce a better output. In the first example in Table 7, the baseline

Description	Sentence
Subject verb agreement	
source	Unlike people with high proportion of skin pigment , **people** with albinism **is** prone to sunburn .
baseline	Unlike people with high proportion of skin pigment , **people** with albinism **is** prone to sunburn .
edit selection	Unlike people with high proportion of skin pigment , **people** with albinism **are** prone to sunburn .
reference	Unlike people with high proportion of skin pigment , **people** with albinism **are** prone to sunburn .
Article noun agreement	
source	As a result , one does not train the necessary parts of the skills required for **a** proper interpersonal **relations** .
baseline	As a result , one does not train the necessary parts of the skills required for **a** proper interpersonal **relations** .
edit selection	As a result , one does not train the necessary parts of the skills required for proper interpersonal **relations** .
reference	As a result , one does not practise the necessary skills required for proper interpersonal **relations** .
Verb preposition agreement	
source	Some people **spend** a lot of time **in** it and forget their real life .
baseline	Some people **spend** a lot of time **in** it and forget their real life .
edit selection	Some people **spend** a lot of time **on** it and forget their real life .
reference	Some people **spend** a lot of time **on** them and forget their real life .

Table 7: Example output after edit selection on the 5-best hypotheses compared to the baseline SMT system output. *source* refers to the original erroneous sentence written by an ESL learner. *reference* is the corrected sentence by a human annotator.

system is not able to correct the verb *is* to *are*. However, the edit selection method considers valid edits from other hypotheses to give a better correction. For example, the nearest noun phrase head to the left of the edit, *people*, helps the classifier gain context knowledge. The edit $is \rightarrow are$ is evaluated as a valid edit.

Similarly, there are other types of errors which are difficult for the SMT approach, but can be overcome by the use of the edit classifier. The second example shows the improper usage of the article *a* for the plural noun *relations*. The interaction between the article and the head noun is not captured by the baseline system because of the distance between the two words in the sentence. The nearest noun phrase head to the right of the edit, which is a feature in our classifier, can capture the dependency between the article and its head noun. The third example also shows a similar weakness of the baseline system when it fails to correct the preposition error. The nearest verb phrase head to the left of the edit, a feature in the edit classifier, helps to correct the preposition error.

In addition, we note that edit selection can be seen as a general method to combine multiple correction candidates. Unlike the reranking method, this method can be applied to hypotheses generated by multiple GEC systems.

7   Conclusion

In this paper, we have presented our novel approach to improve the accuracy of grammatical error correction, by exploiting the n-best hypotheses of an SMT-based GEC system. Our methods further improve the performance of the SMT approach to GEC, which is one of the dominant approaches for GEC.

We first classify the corrections made by a baseline SMT system as valid or invalid using supervised learning. An edit classifier is built to distinguish between valid versus invalid edits made by the system. We then introduce two novel methods which exploit the list of n-best hypotheses using the edit classifier. The first method is to integrate the classifier's scores into a reranking method. The second method is to select a non-overlapping subset of valid edits which attains the

best score. On the CoNLL-2014 test set, both methods give significant improvements compared to a competitive baseline. Our best method achieves an $F_{0.5}$ score of 41.19% by selecting edits from the list of 5-best hypotheses.

Apart from the hypothesis rank feature in our classifier, all other features of the classifier are independent of the baseline SMT system. This opens up the possibility of using our classifier to combine various correction candidates from multiple GEC systems as well.

Acknowledgments

References

[Brockett *et al.*, 2006]   Chris Brockett, William B Dolan, and Michael Gamon.   Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[Dahlmeier and Ng, 2012]   Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012.

[Dahlmeier *et al.*, 2013]   Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu.   Building a large annotated corpus of learner English: The NUS corpus of learner English.   In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, 2013.

[Dredze *et al.*, 2008]   Mark Dredze, Koby Crammer, and Fernando Pereira.   Confidence-weighted linear classification. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[Heafield and Lavie, 2010] Kenneth Heafield and Alon Lavie. Combining machine translation output with open source: The Carnegie Mellon multi-engine machine translation scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36, 2010.

[Heafield, 2011] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011.

[Junczys-Dowmunt and Grundkiewicz, 2014] Marcin Junczys-Dowmunt and Roman Grundkiewicz. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[Koehn et al., 2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Interactive Poster and Demonstration Session*, 2007.

[Mizumoto and Matsumoto, 2016] Tomoya Mizumoto and Yuji Matsumoto. Discriminative reranking for grammatical error correction with statistical machine translation. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

[Mizumoto et al., 2011] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, 2011.

[Ng et al., 2013] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 2013.

[Ng et al., 2014] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014.

[Och and Ney, 2002] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[Och and Ney, 2003] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[Och, 2003] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003.

[Rozovskaya et al., 2014] Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. The Illinois-Columbia system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014.

[Susanto et al., 2014] Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[Tajiri et al., 2012] Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. Tense and aspect error correction for ESL learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012.

[Yoshimoto et al., 2013] Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, and Yuji Matsumoto. NAIST at 2013 CoNLL grammatical error correction shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 2013.

[Yuan and Briscoe, 2016] Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2016.

[Yuan and Felice, 2013] Zheng Yuan and Mariano Felice. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 2013.

[Yuan et al., 2016] Zheng Yuan, Ted Briscoe, and Mariano Felice. Candidate re-ranking for SMT-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, 2016.