Connecting the Dots: Towards Human-Level Grammatical Error Correction

Shamil Chollampatt[1] and **Hwee Tou Ng**[1,2]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]Department of Computer Science
National University of Singapore
shamil@u.nus.edu, nght@comp.nus.edu.sg

Abstract

We build a grammatical error correction (GEC) system primarily based on the state-of-the-art statistical machine translation (SMT) approach, using task-specific features and tuning, and further enhance it with the modeling power of neural network joint models. The SMT-based system is weak in generalizing beyond patterns seen during training and lacks granularity below the word level. To address this issue, we incorporate a character-level SMT component targeting the misspelled words that the original SMT-based system fails to correct. Our final system achieves 53.14% $F_{0.5}$ score on the benchmark CoNLL-2014 test set, an improvement of 3.62% $F_{0.5}$ over the best previous published score.

1 Introduction

Grammatical error correction (GEC) is the task of correcting various textual errors including spelling, grammar, and collocation errors. The phrase-based statistical machine translation (SMT) approach is able to achieve state-of-the-art performance on GEC (Junczys-Dowmunt and Grundkiewicz, 2016). In this approach, error correction is treated as a machine translation task from the language of "bad English" to the language of "good English". SMT-based systems do not rely on language-specific tools and hence they can be trained for any language with adequate parallel data (i.e., erroneous and corrected sentence pairs). They are also capable of correcting complex errors which are difficult for classifier systems that target specific error types. The generalization of SMT-based GEC systems has been shown to improve further by adding neural network models (Chollampatt et al., 2016b).

Though SMT provides a strong framework for GEC, the traditional word-level SMT is weak in generalizing beyond patterns seen in the training data (Susanto et al., 2014; Rozovskaya and Roth, 2016). This effect is particularly evident for spelling errors, since a large number of misspelled words produced by learners are not observed in the training data. We propose improving the SMT approach by adding a character-level SMT component to a word-level SMT-based GEC system, with the aim of correcting misspelled words.

Our word-level SMT-based GEC system utilizes task-specific features described in (Junczys-Dowmunt and Grundkiewicz, 2016). We show in this paper that performance continues to improve further after adding neural network joint models (NNJMs), as introduced in (Chollampatt et al., 2016b). NNJMs can leverage the continuous space representation of words and phrases and can capture a larger context from the source sentence, which enables them to make better predictions than traditional language models (Devlin et al., 2014). The NNJM is further improved using the regularized adaptive training method described in (Chollampatt et al., 2016a) on a higher quality training dataset, which has a higher error-per-sentence ratio. In addition, we add a character-level SMT component to generate candidate corrections for misspelled words. These candidate corrections are rescored with n-gram language model features to prune away non-word candidates and select the candidate that best fits the context. Our final system outperforms the best prior published system when evaluated on the benchmark CoNLL-2014 test set. For better replicability, we release our source code and model files publicly at https://github.com/nusnlp/smtgec2017.

Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 327–333
Copenhagen, Denmark, September 8, 2017.

2 Related Work

GEC has gained popularity since the CoNLL-2014 (Ng et al., 2014) shared task was organized. Unlike previous shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013) that focused only on a few error types, the CoNLL-2014 shared task dealt with correction of all kinds of textual errors. The SMT approach, which was first used for correcting countability errors of mass nouns (Brockett et al., 2006), became popular during the CoNLL-2014 shared task. Two of the top three teams used this approach in their systems. It later became the most widely used approach and was used in state-of-the-art GEC systems (Susanto et al., 2014; Chollampatt et al., 2016b; Junczys-Dowmunt and Grundkiewicz, 2016; Rozovskaya and Roth, 2016). Neural machine translation approaches have also showed some promise (Xie et al., 2016; Yuan and Briscoe, 2016).

A number of papers on GEC were published in 2016. Chollampatt et al. (2016b) showed that using neural network translation models in phrase-based SMT decoding improves performance. Other works focused on re-ranking and combination of the n-best hypotheses produced by an SMT system using classifiers to generate better corrections (Mizumoto and Matsumoto, 2016; Yuan et al., 2016; Hoang et al., 2016). Rozovskaya and Roth (2016) compared the SMT and classifier approaches by performing error analysis of outputs and described a pipeline system using classifier-based error type-specific components, a context sensitive spelling correction system (Flor and Futagi, 2012), punctuation and casing correction systems, and SMT. Junczys-Dowmunt and Grundkiewicz (2016) described a state-of-the-art SMT-based GEC system using task-specific features, better language models, and task-specific tuning of the SMT system. Their system achieved the best published score to date on the CoNLL-2014 test set. We use the features proposed in their work to enhance the SMT component in our system as well. Additionally, we use neural network joint models (Devlin et al., 2014) introduced in (Chollampatt et al., 2016b) and a character-level SMT component.

Character-level SMT systems are used in transliteration and machine translation (Tiedemann, 2009; Nakov and Tiedemann, 2012; Durrani et al., 2014). It has been previously used for spelling correction in Arabic (Bougares and Bouamor, 2015) and for pre-processing noisy input to an SMT system (Formiga and Fonollosa, 2012).

3 Statistical Machine Translation

We use the popular phrase-based SMT toolkit Moses (Koehn et al., 2007), which employs a log-linear model for combination of features. We use the task-specific tuning and features proposed in (Junczys-Dowmunt and Grundkiewicz, 2016) to further improve the system. The features include edit operation counts, a word class language model (WCLM), the Operation Sequence Model (OSM) (Durrani et al., 2013), and sparse edit operations. Moreover, Junczys-Dowmunt and Grundkiewicz (2016) trained a web-scale language model (LM) using large corpora from the Common Crawl data (Buck et al., 2014). We train an LM of similar size from the same corpora and use it to improve our GEC performance.

4 Neural Network Joint Models and Adaptation

Following Chollampatt et al. (2016b), we add a neural network joint model (NNJM) feature to further improve the SMT component. We train the neural networks on GPUs using log-likelihood objective function with self-normalization, following (Devlin et al., 2014). Training of the neural network joint model is done using a Theano-based (Theano Development Team, 2016) implementation, CoreLM[1]. Chollampatt et al. (2016a) proposed adapting SMT-based GEC based on the native language of writers, by adaptive training of a pre-trained NNJM on in-domain data (written by authors sharing the same native language) using a regularized loss function. We follow this adaptation method and perform subsequent adaptive training of the NNJM, but on a subset of training data with better annotation quality and a higher error-per-sentence ratio, favoring more corrections and thus increasing recall.

5 Spelling Error Correction using SMT

Due to the inherent weakness of SMT-based GEC systems in correcting unknown words (mainly consisting of misspelled words), we add a character-level SMT component for spelling error correction. A character in this character-level

[1] https://github.com/nusnlp/corelm

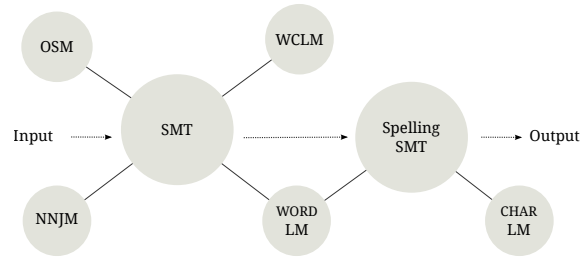

Figure 1: Architecture of our complete SMT-based system.

SMT component is equivalent to a word in word-level SMT, and a sequence of characters (i.e., a word) in the former is equivalent to a sequence of words (i.e., a sentence) in the latter. Input to our character-level SMT component is a sequence of characters that make up the unknown (misspelled) word and output is a list of correction candidates (words). Note that unknown words are words unseen in the source side of the parallel training data used to train the translation model. For training the character-level SMT component, alignments are computed based on a Levenshtein matrix, instead of using GIZA++ (Och and Ney, 2003). Our character-level SMT is tuned using the M^2 metric (Dahlmeier and Ng, 2012) on characters, with character-level edit operation features and a 5-gram character LM. For each unknown word, character-level SMT produces 100 candidates that are then rescored to select the best candidate based on the context. This rescoring is done following Durrani et al. (2014) and uses word-level n-gram LM features: LM probability and the LM OOV (out-of-vocabulary) count denoting the number of words in the sentence that are not in the LM's vocabulary. The architecture of our final system is shown in Figure 1.

6 Experiments

6.1 Data and Evaluation

The parallel data for training our word-level SMT system consist of two corpora: the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) and Lang-8 Learner Corpora v2 (Lang-8) (Mizumoto et al., 2011). From NUCLE, we extract sentences with at least one annotation (edit) in a sentence. We use one-fourth of these sentences as our development data (5,458 sentences with 141,978 source tokens). The remainder of NUCLE, including sentences without annotations (i.e., error-free sentences), are used for training. We extract the English portion of Lang-8 by selecting sentences written by English learners via filtering using a language identification tool, langid.py (Lui and Baldwin, 2012). This filtered data set and the training portion of NUCLE are combined to form the training set, consisting of 2.21M sentences (26.77M source tokens and 30.87M target tokens). We use two corpora to train the LMs: Wikipedia texts (1.78B tokens) and a subset of the Common Crawl corpus (94B tokens). To train the character-level SMT component, we obtain a corpus of misspelled words and their corrections[2], of which the misspelling-correction pairs from Holbrook are used as the development set and the remaining pairs together with the unique words in the NUCLE training data (replicated on the source side to get parallel data) are used for training.

We evaluate our system on the official CoNLL-2014 test set, using the MaxMatch (Dahlmeier and Ng, 2012) scorer v3.2 which computes the $F_{0.5}$ score, as well as on the JFLEG corpus (Napoles et al., 2017), an error-corrected subset of the GUG corpus (Heilman et al., 2014), using the $F_{0.5}$ and GLEU (Napoles et al., 2015) metrics.

6.2 SMT-Based GEC System

Our SMT-based GEC system uses a phrase table trained on the complete parallel data. In our word-level SMT system, we use two 5-gram LMs, one of them trained on the target side of the parallel training data and the other trained on Wikipedia texts (Wiki LM). We add all the *dense* features proposed in (Junczys-Dowmunt and Grundkiewicz, 2016) and *sparse* edit features on words (with one word context). We further improve the system by replacing Wiki LM with a 5-gram LM trained on Common Crawl data (94BCC LM). NNJM is trained on the complete parallel data. We further adapt the NNJM following the adaptation method proposed by Chollampatt et al. (2016a) on sentences from the training portion of NUCLE that contain at least one error annotation (edit) in a sentence. We use the same hyper-parameters as (Chollampatt et al., 2016a). The SMT-based GEC system with all the features, 94BCC LM, and adapted NNJM, is referred to as "Word SMT-GEC".

[2]http://www.dcs.bbk.ac.uk/~ROGER/corpora.html

System	CoNLL-2014		
	Prec.	Recall	$F_{0.5}$
SMT-GEC	55.96	22.54	43.16
+ dense + sparse features	58.24	24.84	45.90
– Wiki LM + 94BCC LM	61.02	27.80	49.25
+ NNJM	61.65	29.11	50.39
+ adaptation	62.14	30.92	51.70
[Word SMT-GEC]			
+ Spelling SMT	**62.74**	**32.96**	**53.14**
[Word&Char SMT-GEC]			

Table 1: Results of incremental addition of features and components.

System	Official Test ($F_{0.5}$)	Bryant and Ng (2015)		
		10 ann. ($F_{0.5}$)	SvH ($F_{0.5}$)	Ratio (%)
Word SMT-GEC	51.70	68.38	67.51	93.02
Word&Char SMT-GEC	**53.14**	**69.12**	**68.29**	**94.09**
J&G (2016)	49.52	66.83	65.90	90.79
R&R (2016)	47.40	62.45	61.50	84.73
CoNLL-2014 Top System				
Felice et al. (2014)	37.33	54.30	53.47	73.67

Table 2: Comparison on the CoNLL-2014 test set.

System	Dev		Test	
	$F_{0.5}$	GLEU	$F_{0.5}$	GLEU
Word SMT-GEC	58.17	48.17	60.95	53.18
Word&Char SMT-GEC	**61.51**	**51.01**	**64.25**	**56.78**
Yuan and Briscoe (2016)	50.8	47.20	–	52.05
Chollampatt et al. (2016a)	52.7	46.27	–	50.13

Table 3: Results on the JFLEG corpus.

6.3 SMT for Spelling Error Correction

The character-level SMT component that generates candidates for misspelled words uses a 5-gram character-level LM trained on the target side of the spelling corpora. 5-gram Wiki LM is used during rescoring. The final system is referred to as "Word&Char SMT-GEC".

7 Results and Discussions

Table 1 shows the results of incrementally adding features and components to the SMT-GEC system, measuring performance on the official CoNLL-2014 test set. All SMT systems are tuned five times and the feature weights are averaged in order to account for optimizer instability. The improvement obtained for each incremental modification is statistically significant ($p < 0.01$) over its previous system.

The addition of NNJM improves by 1.14% $F_{0.5}$ on top of a high-performing SMT-based GEC system with task-specific features and a web-scale LM. Adaptation of NNJM on a subset of NUCLE improves the results by a notable margin (1.31% $F_{0.5}$). The NUCLE data set is manually annotated by experts and is of higher quality than Lang-8 data. Also, choosing sentences with a higher error rate encourages NNJM to favor more corrections.

Adding the SMT component for spelling error correction ("Spelling SMT") further improves $F_{0.5}$ to 53.14%. We use Wiki LM to rescore the candidates, since using 94BCC LM yielded slightly worse results (53.06% $F_{0.5}$). 94BCC LM, trained on noisy web texts, includes many misspellings in its vocabulary and hence misspelled translation candidates are not effectively pruned away by the OOV feature compared to using Wiki LM.

7.1 Comparison to the State of the Art

Table 2 shows the comparison of our systems to other top-performing systems: Junczys-Dowmunt and Grundkiewicz (2016) (J&G) and Rozovskaya and Roth (2016) (R&R)[3]. "Word SMT-GEC" is better than the previous best system (J&G) by a margin of 2.18% $F_{0.5}$. This improvement is without using any additional datasets compared to J&G. "Word&Char SMT-GEC", which additionally uses "Spelling SMT" trained using spelling corpora, increases the margin of improvement to 3.62% $F_{0.5}$ and becomes the new state of the art.

We also evaluate using 10 sets of human annotations of the CoNLL-2014 test set released by Bryant and Ng (2015) ("10 ann."). We measure a system's performance compared to human using the ratio metric ("Ratio"), which is the average system-vs-human score ("SvH") divided by average human-vs-human score ($F_{0.5}$ of 72.58%). "SvH" is computed by removing one set of human annotations at a time and evaluating the system against the remaining 9 sets, and finally averaging over all 10 repetitions. The results show that "Word&Char SMT-GEC" achieves 94.09% of the human-level performance, substantially closing the gap between system and human performance for this task by 36%.

To ascertain the generalizability of our results, we also evaluate our system on the JFLEG development and test sets without re-tuning. Table 3 compares our systems with top-performing systems[4]. Our systems outperform the previous best systems by large margins.

[3]We re-run the official scorer (v3.2) on the released outputs of these systems against the official test set as well as the annotations released by Bryant and Ng (2015).

[4]Results are obtained from (Napoles et al., 2017) and https://github.com/keisks/jfleg

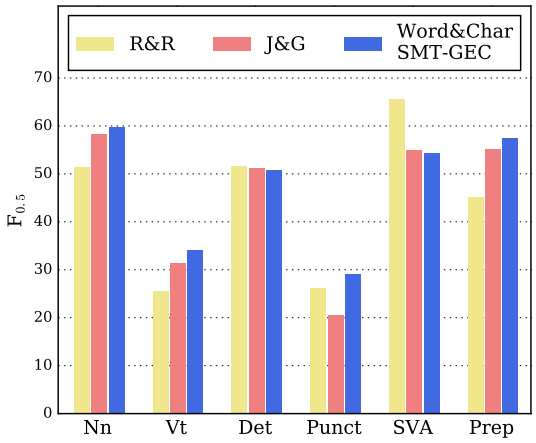

Figure 2: Per-error-type $F_{0.5}$ on CoNLL-2014 test set.

7.2 Error Type Analysis

We analyze the performance of our final system and the top systems on specific error types on the CoNLL-2014 test set. To do this, we compare the per-error-type $F_{0.5}$ using the ERRANT toolkit (Bryant et al., 2017). ERRANT uses a rule-based framework primarily relying on part-of-speech (POS) tags to classify the error types. The error type classification has been shown to achieve 95% acceptance by human raters.

We analyze the performance on six common error types, namely, noun number (*Nn*), verb tense (*Vt*), determiner (*Det*), punctuation (*Punct*), subject-verb agreement (*SVA*), and preposition (*Prep*) errors. The results are shown in Figure 2. Our system outperforms the other systems on four of these six error types, and achieves comparable performance on the determiner errors. It is interesting to note that R&R outperforms our system and J&G on subject-verb agreement errors by a notable margin. This is because R&R uses a classification-based system for subject-verb agreement errors that uses rich linguistic features including syntactic and dependency parse information. SMT-based systems are weaker in correcting such errors as they do not explicitly identify and model the relationship between a verb and its subject.

7.3 Performance on Spelling Errors

We perform comparative analysis on spelling error correction on the CoNLL-2014 test set using ERRANT. The results are summarized in Table 4. Our final system with the character-level SMT component, "Word&Char SMT-GEC", achieves the highest recall (91.35) and $F_{0.5}$ (78.12) compared to the other systems. J&G and 'Word SMT-GEC" rely solely on misspelling-correction patterns seen during training for spelling correction. These two systems achieve the highest precision values (82.35 and 76.36, respectively) but have very low recall values (46.15 and 46.67, respectively) as they do not generalize to unseen misspellings. R&R, on the other hand, uses a specialized context-sensitive spelling error correction component, ConSpel (Flor and Futagi, 2012). ConSpel is a proprietary non-word spell checker that has been shown to outperform off-the-shelf spell checkers such as MS Word and Aspell. Despite using ConSpel, R&R achieves a lower precision (74.19 vs. 75.40) and recall (85.98 vs. 91.35) compared to our final system. We also compare against a baseline where our spelling correction component is replaced by an off-the-shelf spell checker Hunspell ("Word SMT-GEC + Hunspell"). Using Hunspell causes a drastic drop in precision due to a large number of spurious corrections that it proposes and results in a lower $F_{0.5}$ score.

System	Precision	Recall	$F_{0.5}$
J&G (2016)	82.35	46.15	71.19
R&R (2016)	74.19	85.98	76.29
Word SMT-GEC	76.36	46.67	67.74
Word SMT-GEC + Hunspell	58.94	86.41	62.94
Word&Char SMT-GEC	75.40	91.35	78.12

Table 4: Performance on spelling error correction.

8 Conclusion

We have improved a state-of-the-art SMT-based GEC system by incorporating and adapting neural network joint models. The weakness of SMT-based GEC in correcting misspellings is addressed by adding a character-level SMT component. Our final best system achieves 53.14% $F_{0.5}$ on the CoNLL-2014 test set, outperforming the previous best system by 3.62%, and achieves 94% of human performance on this task.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This research was supported by Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2013-T2-1-150.

References

Fethi Bougares and Houda Bouamor. 2015. UMMU@QALB-2015 shared task: Character and word level SMT pipeline for automatic error correction of Arabic text. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*.

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

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the Common Crawl. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng. 2016a. Adapting grammatical error correction based on the native language of writers with neural network joint models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016b. Neural network translation models for grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. 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*.

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 the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can Markov models over minimal translation units help phrase-based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2: Short Papers*.

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 the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

Michael Flor and Yoko Futagi. 2012. On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*.

Lluís Formiga and José A. R. Fonollosa. 2012. Dealing with input noise in statistical machine translation. In *Proceedings of COLING 2012: Posters*.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an SMT approach to grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra

Constantin, and Evan Herbst. 2007. 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 Sessions)*.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*.

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 the Fifth International Joint Conference on Natural Language Processing*.

Tomoya Mizumoto and Yuji Matsumoto. 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Preslav Nakov and Jörg Tiedemann. 2012. Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*.

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 the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

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

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

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

Jörg Tiedemann. 2009. Character-based PSMT for closely related languages. In *Proceedings of the 13th Conference of the European Association for Machine Translation*.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

Zheng Yuan and Ted Briscoe. 2016. 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*.

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