Factored Statistical Machine Translation for Grammatical Error Correction

Yiming Wang, **Longyue Wang, Derek F. Wong, Lidia S. Chao, Xiaodong Zeng, Yi Lu**
Natural Language Processing & Portuguese-Chinese Machine Translation Laboratory,
Department of Computer and Information Science,
University of Macau, Macau S.A.R., China
{wang2008499,vincentwang0229}@gmail.com,derekfw@umac.mo,
lidiasc@umac.mo,nlp2ct.samuel@gmail.com,mb25435@umac.mo

Abstract

This paper describes our ongoing work on grammatical error correction (GEC). Focusing on all possible error types in a real-life environment, we propose a factored statistical machine translation (SMT) model for this task. We consider error correction as a series of language translation problems guided by various linguistic information, as factors that influence translation results. Factors included in our study are morphological information, i.e. word stem, prefix, suffix, and Part-of-Speech (PoS) information. In addition, we also experimented with different combinations of translation models (TM), phrase-based and factor-based, trained on various datasets to boost the overall performance. Empirical results show that the proposed model yields an improvement of 32.54% over a baseline phrase-based SMT model. The system participated in the CoNLL 2014 shared task and achieved the 7th and 5th $F_{0.5}$ scores[1] on the official test set among the thirteen participating teams.

1 Introduction

The task of grammatical error detection and correction (GEC) is to make use of computational methods to fix the mistakes in a written text. It is useful in two aspects. For a non-native English learner it may help to improve the grammatical quality of the written text. For a native speaker the tool may help to remedy mistakes automatically. Automatic correction of grammatical errors is an active research topic, aiming at improving the writing process with the help of artificial intelligent techniques. Second language learning is a user group of particular interest.

Recently, Helping Our Own (HOO) and CoNLL held a number of shared tasks on this topic (Dale et al., 2012, Ng et al., 2013, Ng et al., 2014). Previous studies based on rules (Sidorov et al., 2013), data-driven methods (Berend et al., 2013, Yi et al., 2013) and hybrid methods (Putra and Szabó, 2013, Xing et al., 2013) have shown substantial gains for some frequent error types over baseline methods. Most proposed methods share the commonality that a sub-model is built for a specific type of error, on top of which a strategy is applied to combine a number of these individual models. Also, detection and correction are often split into two steps. For example, Xing et al. (2013) presented the UM-Checker for five error types in the CoNLL 2013 shared task. The system implements a cascade of five individual *detection-and-correction* models for different types of error. Given an input sentence, errors are detected and corrected one-by-one by each sub-model at the level of its corresponding error type. The specifics of an error type are fully considered in each sub-model, which is easier to realize for a single error type than for multiple types in a single model. In addition, dividing the error detection and correction into two steps alleviates the application of machine learning classifiers. However, an approach that considers error types individually may have negative effects:

- This approach assumes independence between each error type. It ignores the interaction of neighboring errors. Results (Xing et al., 2013) have shown that

[1] These two rankings are based on gold-standard edits without and with alternative answers, respectively.

Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pages 83–90,
Baltimore, Maryland, 26-27 July 2014.

consecutive errors of multiple types tend to hinder solving these errors individually.

- As the number of error types increases, the complexities of analyzing, designing, and implementing the model increase, in particular when combinatorial errors are taken into account.
- Looking for an optimal model combination becomes complex. A simple pipeline approach would result in interference and the generation of new errors, and hence to propagating those errors to the subsequent processes.
- Separating the detection and correction tasks may result in more errors. For instance, once a candidate is misidentified as an error, it would be further revised and turned into an error by the correction model. In this scenario the model risks losing precision.

In the shared task of this year (Ng et la., 2014), two novelties are introduced: 1) all types of errors present in an essay are to be detected and corrected (i.e., there is no restriction on the five error types of the 2013 shared task); 2) the official evaluation metric of this year adopts $F_{0.5}$, weighting precision twice as much as recall. This requires us to explore an alternative universal joint model that can tackle various kinds of grammatical errors as well as join the detection and correction processes together. Regarding grammatical error correction as a process of translation has been shown to be effective (Ehsan and Faili, 2013, Mizumoto et al., 2011, Yoshimoto et al., 2013, Yuan and Felice, 2013). We treat the problematic sentences and golden sentences as pairs of source and target sentences. In SMT, a translation model is trained on a parallel corpus that consists of the source sentences (i.e. sentences that may contain grammatical errors) and the targeted translations (i.e. the grammatically well-formed sentences). The challenge is that we need a large amount of these parallel sentences for constructing such a data-driven SMT system. Some researches (Brockett et al., 2006, Yuan and Felice, 2013) explore generating artificial errors to resolve this sparsity problem. Other studies (Ehsan and Faili, 2013, Yoshimoto et al., 2013, Yuan and Felice, 2013) focus on using syntactic information (such as PoS or tree structure) to enhance the SMT models.

In this paper, we propose a factored SMT model by taking into account not only the surface information contained in the sentence, but also morphological and syntactic clues (i.e., word stem, prefix, suffix and finer PoS information). To counter the sparsity problem we do not use artificial or manual approaches to enrich the training data. Instead we apply factored and transductive learning techniques to enhance the model on a small dataset. In addition, we also experimented with different combinations of translation models (TM), phrase- and factor-based, that are trained on different datasets to boost the overall performance. Empirical results show that the proposed model yields an improvement of 32.54% over a baseline phrase-based SMT model.

The remainder of this paper is organized as follows: Section 2 describes our proposed methods. Section 3 reports on the design of our experiments. We discuss the result, including the official shared task results, in Section 4,. We summarize our conclusions in Section 5.

2 Methodology

In contrast with phrase-based translation models, factored models make use of additional linguistic clues to guide the system such that it generates translated sentences in which morphological and syntactic constraints are met (Koehn and Hoang, 2007). The linguistic clues are taken as factors in a factored model; words are represented as vectors of factors rather than as a single token. This requires us to pre-process the training data to factorize all words. In this study, we explore the use of various types of morphological information and PoS as factors. For each possible factor we build an individual translation model. The effectiveness of all factors is analyzed by comparing the performance of the corresponding models on the grammatical error correction task. Furthermore, two approaches are proposed to combine those models. One adopts the model cascading method based on transductive learning. The second approach relies on learning and decoding multiple factors learning. The details of each approach are discussed in the following sub-sections.

2.1 Data Preparation

In order to construct a SMT model, we convert the training data into a parallel corpus where the problematic sentences that ought to be corrected are regarded as source sentences, while the reference sentences are treated as the corresponding target translations. We discovered that a number of sentences is absent at the target side due to incorrect annotations in the golden

data. We removed these unparalleled sentences from the data. Secondly, the initial capitalizations of sentences are converted to their most probable casing using the Moses *truecaser*[2]. URLs are quite common in the corpus, but they are not useful for learning and even may cause the model to apply unnecessary correction on it. Thus, we mark all of the ULRs with XML markups, signaling the SMT decoder not to analyze an URL and output it as is.

2.2 Model Construction

In this study we explore four different factors: prefix, suffix, stem, and PoS. This linguistic information not only helps to capture the local constraints of word morphologies and the interaction of adjacent words, but also helps to prevent data sparsity caused by inflected word variants and insufficient training data.

Word stem: Instead of lemmas, we prefer word stemming as one of the factors, considering that stemming does not requires deep morphological analysis and is easier to obtain. Second, during the whole error detection and correction process, stemming information is used as auxiliary information in addition to the original word form. Third, for grammatical error correction using word lemmas or word stems in factored translation model shows no significant difference. This is because we are translating text of the same language, and the translation of this factor, stem or lemma, is straightforwardly captured by the model. Hence, we do not rely on the word lemma. In this work, we use the English Porter stemmer (Porter, 1980) for generating word stems.

Prefix: The second type of morphological information we explored is the word prefix. Although a prefix does not present strong evidence to be useful to the grammatical error correction, we include it in our study in order to fully investigate all types of morphological information. We believe the prefix can be an important factor in the correction of initial capitalization, e.g. "*In this era*, engineering *designs…*" should be changed to "*In this era*, engineering *designs…*" In model construction, we take the first three letters of a word as its prefix. If the length of a word is less than three, we use the word as the prefix factor.

Suffix: Suffix, one of the important factors, helps to capture the grammatical agreements between predicates and arguments within a sentence. Particularly the endings of plural nouns and inflected verb variants are useful for the detection of agreement violations that shown up in word morphologies. Similar to how we represent the prefix, we are interested in the last three characters of a word.

	Examples
Sentence	this card contains biometric data *to* add security and reduce *the* risk *of* falsification
Original POS	DT NN BVZ JJ NNS **TO** VB NN CC VB **DT** NN **IN** NN
Specific POS	DT NN VBZ JJ NNS **TO_to** VB NN CC VB **DT_the** NN **IN_of** NN

Table 1: Example of modified PoS.

According to the description of factors, Figure 1 illustrates the forms of various factors extracted from a given example sentence.

Surface	*constantly combining ideas will result in better solutions being formulated*
Prefix	*con com ide wil res in bet sol bei for*
Suffix	*tly ing eas ill ult in ter ons ing ted*
Stem	*constantli combin idea will result in better solut be formul*
Specific POS	RB VBG NNS MD VB IN JJR NNS VBG VBN

Figure 1: The factorized sentence.

PoS: Part-of-Speech tags denote the morpho-syntactic category of a word. The use of PoS sequences enables us to some extent to recover missing determiners, articles, prepositions, as well as the modal verb in a sentence. Empirical studies (Yuan and Felice, 2013) have demonstrated that the use of this information can greatly improve the accuracy of the grammatical error correction. To obtain the PoS, we adopt the Penn Treebank tag set (Marcus et al., 1993), which contains 45 PoS tags. The Stanford parser (Klein and Manning, 2002) is used to extract the PoS information. Inspired by Yuan and Felice (2013), who used preposition-specific tags to fix the problem of being unable to distinguish between prepositions and obtained good performance, we create specific tags both for determiners (i.e., *a*, *an*, *the*) and prepositions. Table 1 provides an example of this modification, where prepositions, **TO** and **IN**, and determiner,

[2] After decoding, we will de-truecase all these words.

DT, are revised to **TO_to**, **IN_of** and **DT_the,** respectively.

2.3 Model Combination

In addition to the design of different factored translation models, two model combination strategies are designed to treat grammatical error correction problem as a series of translation processes, where an incorrect sentence is translated into the correct one. In both approaches we pipeline two translation models, tm^1 and tm^2. In the first approach, we derive four combinations of different models that trained on different sources.

- In case I, tm_f^1 and tm_f^2 are both factored models but trained on different factors, e.g. for $tm_{f_i}^1$ training on "*surface* + *factor*$_i$" and $tm_{f_j}^2$ on "*surface* + *factor*$_{i \neq j}$". Both models use the same training sentences, but different factors.
- In case II, $tm_{f_j}^2$ is trained on sentences that paired with the output from the previous model, $tm_{f_i}^1$, and the golden correct sentences. We want to create a second model that can also tackle the new errors introduced by the first model.
- In case III, similar to case II, the second translation model, tm_p^2 is replaced by a phrase-based translation model.
- In case IV, the quality of training data is considered vital to the construction of a good translation model. The present training dataset is not large enough. To complement this, the second model, $tm_{f_j}^2$, is trained on an enlarged data set, by combining the training data of both models, i.e. the original parallel data (official incorrect and correct sentence pairs) and the supplementary parallel data (sentences output from the first model, $tm_{f_i}^1$, and the correct sentences). Note that we do not de-duplicate sentences.

In all cases, the testing process is carried out as follows. The test set is translated by the first translation model, $tm_{f_i}^1$. The output from the first model is then fed into the second translation model, $tm_{f_j}^2$. The output of the second model is used as the final corrections.

The second combination approach is to make use of multiple factors for model construction. The question is whether multiple factors when used together may improve the correction results. In this setting we combine two factors together with the word surface form to build a multi-factored translation model. All pairs of factors are used, e.g. stem and PoS. The decoding sequence is as follows: translate the input stems into target stems; translate the PoS; and generate the surface form given the factors of stem and PoS.

3 Experiment Setup

3.1 Dataset

We pre-process the NUCLE corpus (Dahlmeier et al., 2013) as described in Section 2 for training different translation models. We use both the official golden sentences and additional WMT2014 English monolingual data[3] to train an in-domain and a general-domain language model (LM), respectively. These language models are linearly interpolated in the decoding phase. We also randomly select a number of sentence pairs from the parallel corpus as a development set and a test set, disjoint from the training data. Table 2 summarizes the statistics of all the datasets.

Corpus	Sentences	Tokens
Parallel Corpus	55,503	1,124,521 / 1,114,040
Additional Monolingual	85,254,788	2,033,096,800
Dev. Set	500	10,532 / 10,438
Test Set	900	18,032 / 17,906

Table 2: Statistics of used corpora.

The experiments were carried out with MOSES 1.0[4] (Philipp Koehn et al., 2007). The translation and the re-ordering model utilizes the "*grow-diag-final*" symmetrized word-to-word alignments created with GIZA++[5] (Och and Ney, 2003) and the training scripts of MOSES. A 5-gram LM was trained using the SRILM toolkit[6] (Stolcke et al., 2002), exploiting the improved modified Kneser-Ney smoothing (Kneser and Ney, 1995), and quantizing both probabilities and back-off weights. For the log-linear model training, we take minimum-error-rate training (MERT) method as described in (Och, 2003). The result is evaluated by M^2 Scorer (Dahlmeier and Ng, 2012) computing precision, recall and $F_{0.5}$.

[3] http://www.statmt.org/wmt14/translation-task.html.
[4] http://www.statmt.org/moses/.
[5] http://code.google.com/p/giza-pp/.
[6] http://www.speech.sri.com/projects/srilm/.

In total, one baseline system, five individual systems, and four combination systems are evaluated in this study. The baseline system (**Baseline**) is trained on the words-only corpus using a phrase-based translation model. For the individual systems we adopt the factored translation model that are trained respectively on 1) surface and stem factors (**Sys**$_{+stem}$), 2) surface and suffix factors (**Sys**$_{+suf}$), 3) surface and prefix factors (**Sys**$_{+pref}$), 4) surface and PoS factors (**Sys**$_{+PoS}$), and 5) surface and modified-PoS factors (**Sys**$_{+MPoS}$). The combination systems include: 1) the combination of "factored + phrase-based" and "factored + factored" for models cascading; and 2) the factors of surface, stem and modified-PoS (**Sys**$_{+stem+MPoS}$) are combined for constructing a correction system based on a multi-factor model.

4 Results and Discussions

We report our results in terms of the precision, recall and $F_{0.5}$ obtained by each of the individual models and combined models.

4.1 Individual Model

Table **3** shows the absolute measures for the baseline system, while the other individual models are listed with values relative to the baseline.

Model	Precision	Recall	$F_{0.5}$
Baseline	25.58	3.53	11.37
Sys$_{+stem}$	-14.84	+13.00	+0.18
Sys$_{+suf}$	-14.57	+14.77	+0.60
Sys$_{+pref}$	-15.74	+12.20	-0.77
Sys$_{+PoS}$	-11.63	+9.79	+2.45
Sys$_{+MPoS}$	-10.25	+10.60	+3.70

Table 3: Performance of various models.

The baseline system has the highest precision score but the lowest recall. Nearly all individual models except **Sys**$_{+pref}$ show improvements in the correction result ($F_{0.5}$) over the baseline. Overall, **Sys**$_{+MPoS}$ achieves the best result for the grammatical error correction task. It shows a significant improvement over the other models and outperforms the baseline model by 3.7 $F_{0.5}$ score. The **Sys**$_{+stem}$ and **Sys**$_{+suf}$ models obtain an improvement of 0.18 and 0.60 in $F_{0.5}$ scores, respectively, compared to the baseline. Although the differences are not significant, it confirms our hypothesis that morphological clues do help to improve error correction. The $F_{0.5}$ score of **Sys**$_{+pref}$ is the lowest among the models including the baseline, showing a drop of 0.77 in $F_{0.5}$ score against the baseline. One possible reason is that few errors (in the training corpus) involve word prefixes. Thus, the prefix does not seem to be a suitable factor for tackling the GEC problem.

Type	**Sys**$_{+stem}$ (%)	**Sys**$_{+suf}$ (%)	**Sys**$_{+MPoS}$ (%)	Error Num.
Vt	17.07	12.20	12.20	41
ArtOrDet	37.65	36.47	29.41	85
Nn	33.33	19.61	23.53	51
Prep	10.26	10.26	12.82	39
Wci	9.10	10.61	6.10	66
Rloc-	15.20	13.92	10.13	79

Table 4: The capacity of different models in handling six frequent error types.

We analyze the capacities of the models on different types of errors. **Sys**$_{+PoS}$ and **Sys**$_{+MPoS}$ are built by using the PoS and modified PoS. Both of them yield an improvement in $F_{0.5}$ score. Overall, **Sys**$_{+MPoS}$ produces more accurate results than **Sys**$_{+pref}$. Therefore, we specifically compare and evaluate the best three models, **Sys**$_{+stem}$, **Sys**$_{+suf}$ and **Sys**$_{+MPoS}$. Table 4 presents evaluation scores of these models for the six most frequent error types, which take up a large part of the training and test data. Among them, **Sys**$_{+stem}$ displays a powerful capacity to handle determiner and noun/number agreement errors, up to 37.65% and 33.33%. **Sys**$_{+suf}$ shows the ability to correct determiner errors at 36.47%; **Sys**$_{+MPoS}$ yields a similar performance to **Sys**$_{+suf}$. All three individual models exhibit a relatively high capacity to handle determiner errors. The likely reason is that this mistake constitutes the largest portion in training data and test set, giving the learning models many examples to capture this problem well. In the case of preposition errors, **Sys**$_{+MPoS}$ demonstrates a better performance. This, once again, confirms the result (Yuan and Felice, 2013) that the modified PoS factor is effective for every preposition word. For these six error types, the individual models show a weak capacity to handle the word collocation or idiom error category (Wci). Although **Sys**$_{+MPoS}$ achieves the highest $F_{0.5}$ score in the overall evaluation, it only achieves 6.10% in handling this error type. The likely reason is that idioms are not frequent in the training data, and also that in most of the cases they contain out-of-vocabulary words never seen in training data.

4.2 Model Combination

We intend to further boost the overall performance of the correction system by

combining the strengths of individual models through model combination, and compare against the baseline. The systems compared here cover three pipelined models and a multi-factored model, as described earlier in Section 3. The combined systems include: 1) **CSys$_{suf+phrase}$**: the combination of **Sys$_{+suf}$** and the baseline phrase-based translation model; 2) **CSys$_{suf+suf}$**: we combine two similar factored models with suffix factors, **Sys$_{+suf}$**, which is trained on the same corpus; and 3) **TSys$_{suf+phrase}$**: similar to **CSys$_{suf+phrase}$**, but the training data for the second phrase-based model is augmented by adding the output sentences from the previous model (paired with the correct sentences). Our intention is to enlarge the size of the training data. The evaluation results are presented in Table 5.

Model	Precision	Recall	$F_{0.5}$
Baseline	25.58	3.53	11.37
CSys$_{suf+phrase}$	-14.70	+14.61	+0.45
CSys$_{suf+suf}$	-15.04	+14.13	+0.09
TSys$_{suf+phrase}$	-14.76	+14.61	+0.40
Sys$_{+stem+MPoS}$	-15.87	+11.72	-0.90

Table 5: Evaluation results of combined models.

In Table 5 we observe that **Sys$_{+stem+MPoS}$** hurts performance and shows a drop of 0.9% in $F_{0.5}$ score. Both the **CSys$_{suf+phrase}$** and **CSys$_{suf+suf}$** show minor improvements over the baseline system. Even when we enrich the training data for the second model in **TSys$_{suf+phrase}$**, it cannot help in boosting the overall performance of the system. One of the problems we observe is that, with this combination structure, new incorrect sentences are introduced by the model at each step. The errors are propagated and accumulated to the final result. Although **CSys$_{suf+phrase}$** and **CSys$_{suf+suf}$** produce a better $F_{0.5}$ score over the baseline, they are not as good as the individual models, **Sys$_{+PoS}$** and **Sys$_{+MPoS}$**, which are trained on PoS and modified-PoS, respectively.

4.3 The Official Result

After fully evaluating the designed individual models as well as the integrated ones, we adopt **Sys$_{+MPoS}$** as our designated system for this grammatical error correction task. The official test set consists of 50 essays, and 2,203 errors. Table 6 shows the final result obtained by our submitted system.

Table 7 details the correction rate of the five most frequent error types obtained by our system. The result suggests that the proposed system has a better ability in handling the verb, article and determiner error than other error types.

Criteria	Result	Alt. Result
P	0.3127	0.4317
R	0.1446	0.1972
$F_{0.5}$	0.2537	0.3488

Table 6: The official correction results of our submitted system.

Type	Error	Correct	%
Vt	203/201	21/22	10.34/10.94
V0	57/54	9/9	15.79/16.67
Vform	156/169	11/18	7.05/10.65
ArtOrDet	569/656	84/131	14.76/19.97
Nn	319/285	31/42	9.72/10.91

Table 7: Detailed error information of evaluation system (with alternative result).

5 Conclusion

This paper describes our proposed grammatical error detection and correction system based on a factored statistical machine translation approach. We have investigated the effectiveness of models trained with different linguistic information sources, namely morphological clues and syntactic PoS information. In addition, we also explore some ways to combine different models in the system to tackle the correction problem. The constructed models are compared against the baseline model, a phrase-based translation model. Results show that PoS information is a very effective factor, and the model trained with this factor outperforms the others. One difficulty of this year's shared task is that participants have to tackle all 28 types of errors, which is five times more than last year. From the results, it is obvious there are still many rooms for improving the current system.

Acknowledgements

The authors are grateful to the Science and Technology Development Fund of Macau and the Research Committee of the University of Macau for the funding support for their research, under the Reference nos. MYRG076 (Y1-L2)-FST13-WF and MYRG070 (Y1-L2)-FST12-CS. The authors also wish to thank the anonymous reviewers for many helpful comments with special thanks to Antal van den Bosch for his generous help on this manuscript.

References

Gábor Berend, Veronika Vincze, Sina Zarriess, and Richárd Farkas. 2013. LFG-based Features for Noun Number and Article Grammatical Errors. *CoNLL-2013*.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* pages 249–256.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 22-31.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* pages 54–62.

Nava Ehsan, and Heshaam Faili. 2013. Grammatical and context-sensitive error correction using a statistical machine translation framework. *Software: Practice and Experience*. Wiley Online Library.

D. Klein, and C. D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*.

Reinhard Kneser, and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on* Vol. 1, pages 181–184.

P. Koehn, and H. Hoang. 2007. Factored translation models. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* Vol. 868, pages 876–876.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, et al. 2007. Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* pages 177–180.

M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*. MIT Press.

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. *IJCNLP* pages 147–155.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Bryant Christopher. 2014. The conll-2014 shared task on grammatical error correction. *Proceedings of CoNLL*. Baltimore, Maryland, USA.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The conll-2013 shared task on grammatical error correction. *Proceedings of CoNLL*.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation, 160–167.

Franz Josef Och, and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*. MIT Press.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*. MCB UP Ltd.

Desmond Darma Putra, and Lili Szabó. 2013. UdS at the CoNLL 2013 Shared Task. *CoNLL-2013*.

Grigori Sidorov, Anubhav Gupta, Martin Tozer, Dolors Catala, Angels Catena, and Sandrine Fuentes. 2013. Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2). *CoNLL-2013*.

Andreas Stolcke, and others. 2002. SRILM-an extensible language modeling toolkit. *INTERSPEECH*.

Junwen Xing, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Xiaodong Zeng. 2013. UM-Checker: A Hybrid System for English Grammatical Error Correction. *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 34–42. Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W13-3605

Bong-Jun Yi, Ho-Chang Lee, and Hae-Chang Rim. 2013. KUNLP Grammatical Error Correction System For CoNLL-2013 Shared

Task. *CoNLL-2013*.

Ippei Yoshimoto, Tomoya Kose, Kensuke Mitsuzawa, Keisuke Sakaguchi, Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, et al. 2013. NAIST at 2013 CoNLL grammatical error correction shared task. *CoNLL-2013*.

Zheng Yuan, and Mariano Felice. 2013. Constrained grammatical error correction using Statistical Machine Translation. *CoNLL-2013*.