Chinese Grammatical Error Correction Using Statistical and Neural Models

Junpei Zhou[1,2], Chen Li[1(✉)], Hengyou Liu[1], Zuyi Bao[1], Guangwei Xu[1], and Linlin Li[1]

[1] Alibaba Group, 969 West Wenyi Road, Hangzhou, China
{puji.lc,hengyou.lhy,zuyi.bzy,linyan.lll}@alibaba-inc.com,
kunka.xgw@taobao.com
[2] Zhejiang University, 38 Zheda Road, Hangzhou, China
zhoujunpei@zju.edu.cn

Abstract. This paper introduces the Alibaba NLP team's system for NLPCC 2018 shared task of Chinese Grammatical Error Correction (GEC). Chinese as a Second Language (CSL) learners can use this system to correct grammatical errors in texts they wrote. We proposed a method to combine statistical and neural models for the GEC task. This method consists of two modules: the correction module and the combination module. In the correction module, two statistical models and one neural model generate correction candidates for each input sentence. Those two statistical models are a rule-based model and a statistical machine translation (SMT)-based model. The neural model is a neural machine translation (NMT)-based model. In the combination module, we implemented it in a hierarchical manner. We first combined models at a lower level, which means we trained several models with different configurations and combined them. Then we combined those two statistical models and a neural model at the higher level. Our system reached the second place on the leaderboard released by the official.

Keywords: Grammatical Error Correction · Combination
Statistical machine translation · Neural machine translation

1 Introduction

With the economy booming, China becomes more and more attractive to foreign businesses, students, and travelers, and learning Chinese is becoming more and more popular. The number of CSL learners grows up rapidly, but learning Chinese would not be easy for them, because Chinese is quite different from other languages, especially from English. For example, in Chinese, questions are conveyed by intonation and the subject and verb are not inverted as in English. Nouns cannot be post-modified as in English, and adverbials usually precede

J. Zhou—Work done during an internship at Alibaba Group
J. Zhou and C. Li—Equal Contribution.

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11109, pp. 117–128, 2018.
https://doi.org/10.1007/978-3-319-99501-4_10

verbs, unlike in English where complex rules govern the position of such sentence elements. It has quite flexible expressions and loose structural grammar. These traits bring a lot of trouble to CSL learners, leading to the rapid growth of the demands for Chinese GEC.

GEC for English has been studied for many years, with many shared tasks such as CoNLL-2013 [24] and CoNLL-2014 [23], while those kinds of studies on Chinese are less yet. This NLPCC shared task gives researchers an opportunity to build systems and exchange opinions, which can promote progress in this field. Another important contribution of this shared task is that it released a huge dataset for Chinese GEC. The details of this dataset will be described in Sect. 3. This shared task could make the community more flourish which benefits all CSL learners.

This paper is organized as follows: Sect. 2 describes some related works in English as well as Chinese GEC task. Dataset will be described in Sect. 3. Section 4 illustrates our system and explains two modules of it, including three models. The evaluation and discussion of the combination of statistical and neural models are shown in Sect. 5. Section 6 concludes the paper and discusses the future work.

2 Related Work

2.1 English GEC

Earlier methods for English GEC mainly use rule-based approaches [4,13] and classifier-based models [11,25,30], which can correct limited and specific type of errors. To address more complex errors, Machine Translation (MT) models are proposed and developed by many researchers. Statistical Machine Translation (SMT) has been dominant for a long time. In the work of Brockett et al. [2], they propose an SMT GEC model.

Since 2013, the GEC shared tasks in CoNLL2013 [24] and CoNLL2014 [23] boost this field, with a great many approaches developed. A POS-factored SMT system is proposed [34] to correct five types of errors in the text. In the work of Felice et al. [8], they propose a pipeline of the rule-based system and a phrase-based SMT system augmented by a web-based language model. The word-level Levenshtein distance between source and target is used as a translation model feature [15] to enhance the model.

Nevertheless, Neural Machine Translation (NMT) systems have achieved substantial improvements in this field [1,29]. Inspired by this phenomenon, Sun et al. [27] utilize the Convolutional Neural Network (CNN) for the article error correction. The Recurrent Neural Network (RNN) is also used [33] to map the sentence from learner space to expert space.

2.2 Chinese GEC

A great number of resources including annotated corpus are available in English GEC. However, the resource for Chinese is much less, and previous works related to

Chinese GEC is relatively scarce. The NLPTEA CGED shared task [9,16,17,32] boosts the Chinese Grammatical Error Diagnosis (GED) field greatly, and most works in Chinese GEC focus on the detection of errors instead of correction.

A probabilistic first-order inductive learning algorithm [5] outperforms many basic classifiers for error classification. In 2014, Lee et al. propose a judgment system at sentence level [18] combining N-gram statistical features and predefined rules. Several methods including CRF and SVM, together with frequency learning from a large N-gram corpus are used to detect and correct word ordering errors [7]. The work of Chang et al. [6] utilizes rules manually constructed as well as automatically generated. In the work of NTOU [19] they propose a traditional supervised model, which extracts word N-grams and POS N-grams as features. Rule-based methods and n-gram statistical methods are combined [31] to get a hybrid system for the CGED shared task.

3 Dataset Description

The dataset is provided by the NLPCC 2018 GEC shared task. The training data is collected from Lang-8 and each input sentence may have zero to k different corrections. The test data is texts written by foreign students and carefully corrected by professors. Both the training data and the test data are collated into the same form.

Each instance in the training data is in the form of $[s_o, k, \mathbf{C}]$, where s_o is the original sentence written by CSL learners, k denotes the number of correction candidates written by native speakers for s_o, and $\mathbf{C}$ is the set which contains k correction candidates as $\{c_1, c_2, ..., c_k\}$. After thresholding invalid lines whose k is 0, and filtering 216 lines whose $k > 0$ but $\mathbf{C}$ is empty, we got 593,524 valid lines. For each line of the data, we had two options of generating training instances. The first choice is to only use the candidate which has the minimal edit distance from the original sentence. In this method, for an original sentence s_o, we form a training instance (s_o, c_i) where $1 < i < k$ and $c_i = \arg\min_{c_i} EditDistance(c_i, s_o)$. We denote the training set generated by this method as 'NLPCC MinEd', which contains 593,524 data pairs. Another choice is to use all the candidates in $\mathbf{C}$. For an original sentence s_o, we make pairs of the original sentence and each of the candidates to form a training set as $\{(s_o, c_1), (s_o, c_2), ..., (s_o, c_k)\}$. We denote the training set generated by this method as 'NLPCC Expand', which contains 1,097,189 data pairs. This dataset is much larger than previous datasets released for the Chinese GEC field.

4 System Description

Our system combined statistical models as well as neural models, including the correction module and the hierarchical combination module. The pipeline is shown in Fig. 1.

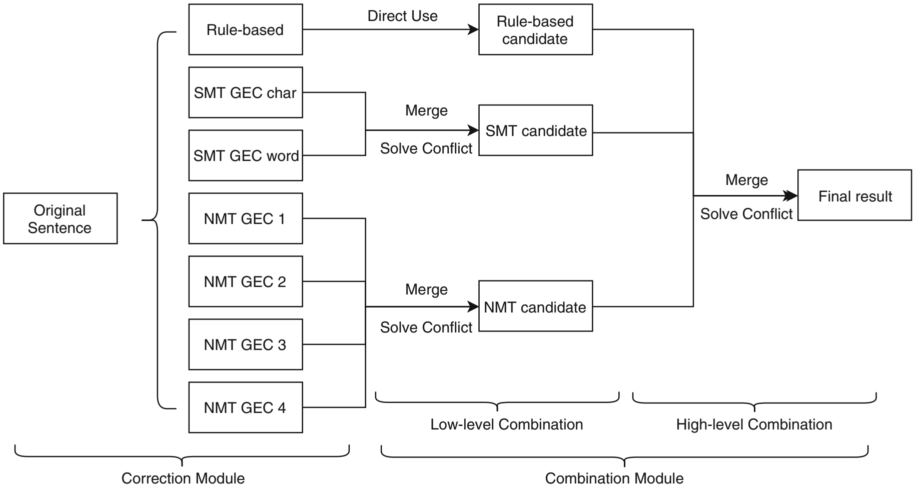

Fig. 1. Pipeline of our system with two modules

4.1 Correction Module

In the correction module, we used both statistical models and neural models with different configurations for the GEC task. The statistical models include the rule-based GEC model and SMT-based GEC models with different configurations. The neural models consist of several NMT-based GEC models with different structures.

Rule-Based GEC Model. The rule-based model starts by segmenting Chinese characters into chunks, which incorporates useful prior grammatical information to identify possible out-of-vocabulary errors. The segments are looked up in the dictionary built by Gigawords [10], and if a segment is out of vocabulary, it will go through the following steps:

1. If the segment consists of two or more characters, and turns out to be in the dictionary by permuting the characters, it will be added to the candidate list.
2. If the concatenation with a previous or next segment is in the dictionary, it will be added to the candidate list.
3. All possible keys in the dictionary with the same or similar Pinyin (the Romanization system for Standard Chinese) or similar strokes to the segment are generated. The generated keys for the segment itself, concatenated with those of previous or next segments, will be added to the candidate list of possible corrections.

After the steps, a candidate list of all possible corrections will be processed to identify whether there might be out-of-vocabulary error and its probability using a language model. The negative log likelihood of a size-5 sliding window suggests whether the top-scored candidate should be a correction of the original segment.

SMT-based GEC Model. The SMT GEC model consists of two components. One is a language model, which assigns a probability $p(e)$ for any target sentence e, and the other one is a translation model, which assigns a conditional probability $p(f|e)$. The language model is learned from a monolingual corpus of the target language, while the parameters of the translation model are calculated from the parallel corpus. We used the noisy channel model [3] to combine the language model and the translation model, and incorporated beam search to decode the result.

To explore the ability of SMT GEC models with different configurations, we trained two SMT GEC models with different data granularity as described in Sect. 5.1, including a char-level model S_{char} and a word-level model S_{word}. The correction result of sentence s_i generated by S_m was denoted as C_{iS_m} where $m \in \{char, word\}$.

NMT-based GEC Model. We used the encoder-decoder structure [1] with the general attention mechanism [20]. The NMT GEC model can capture complex relationships between the original sentence and the corrected sentence in GEC. We used a two-layer LSTM model for both encoder and decoder. To enhance the ability of NMT GEC models, we trained four NMT GEC models with different data pairs and configurations as described in Sect. 5.1. Those four NMT models were denoted as N_j, where $j \in \{1, 2, 3, 4\}$ was the model index. The correction result of sentence s_i generated by N_j was denoted as C_{iN_j}.

We used the character-based NMT because most characters in Chinese have their own meanings, which is quite different from English characters, and the Chinese word's meaning often depends on the meaning of its characters. On the other hand, the errors in original sentences can make the word-based tokenization worse, which will introduce larger and lower quality vocabulary list.

4.2 Combination Module

We performed a hierarchical combination of the correction candidates generated by models in the correction module. The hierarchical combination was composed of a low-level combination and a high-level combination. The low-level combination was used within each category of models, such as combining two SMT GEC models, combining four NMT GEC models, and so on. The high-level combination aimed to combine the candidates generated by the low-level combination, which means that it merged statistical and neural models.

One of the most significant problems in combination is to solve conflicts. The conflict means that when we want to merge several models, a sentence has different candidates from two or more models. For the conflict between two models, we designed five methods to solve the conflict, denoted as M_t where $t \in \{0, 1, 2, 3, 4\}$. M_0 is the simplest method to solve conflicts, which picked one side as the prior side, and then always chose the candidate in the prior side if conflicts occur. M_1 took the union operation on editing sets of two models if conflict occurred. Here the editing set is generated by the difference between the original sentence and the corresponding candidate sentence. Because the editing set is not unique between two sentences, we chose to use the editing set which could minimize the editing distance between two sentences. M_2 took the intersection operation on editing sets of two models if the conflict occurred. M_3 and M_4 both used the language model to assess the quality of candidate sentences. The language model was implemented by *KenLM* [12] to score each of the candidates and picked up the one with the higher score. The only difference is that M_4 used the length of the sentence to normalize the score, which means we divided the score by the length of the candidate sentence.

Low-Level Combination. For the two SMT GEC models S_{char} and S_{word}, we used M_3 to solve the conflict. For the four NMT GEC models, we first picked up two models, because incorporating too many models would confuse the model and provide many wrong candidates. Then we used those M_t where $t \in \{1, 2, 3, 4\}$ methods to solve conflicts between two models. We ranked those four models by the score on the development dataset split from the training dataset. Then we explored the combination of those four models with method M_0, and the order of combination is decided by the ranking order. For example, if model N_3 ranked prior to N_1, it will be used as higher priority during combination, which means that if a sentence s_i has different candidates in C_{iN_3} and C_{iN_1}, we picked up C_{iN_3} as the final result. Following this rule, we found that N_3 combined with N_4 with M_0 performed best, so we used this combination as the backbone, and tested M_t where $t \in \{1, 2, 3, 4\}$ on this combination. The detailed experimental results of the combination can be found in Sect. 5.2.

High-Level Combination. After the low-level combination, for each original sentence s_i, we had three candidates $\{C_{iR}, C_{iS}, C_{iN}\}$ generated by rule-based model, SMT GEC model, and NMT GEC model separately. We performed high-level combination on these candidates. If there were only two candidates which conflicted with each other, we could still use the method described as M_t where $t \in \{0, 1, 2, 3, 4\}$, but when all three candidates conflicted at the same time, we expanded the method M_t to fit three candidates, Those operations in M_t such as union and intersection could be easily expanded. We also designed a protection mechanism for the high-level combination according to the degree of agreement of three GEC models. If those three candidates of GEC models conflicted a lot, we assumed none of them is right and protected the sentence to keep it untouched. According to the sensitivity of the trigger of the protection mechanism, we designed two degrees of protection denoted as P_1 and P_2. The detailed experimental results of the combination can be found in Sect. 5.2.

5 Evaluation and Discussion

5.1 Experimental Settings

We randomly picked 10% of the training dataset as the development dataset, on which we tested our models and chose the combination method according to the scores. Because the official scorer was not released during the contest, we used the script written by ourselves to score the result on the development dataset. Firstly we converted the candidate to the editing set in the form of 'm2', which contains the editing steps between the original sentence and the candidate. Then we also converted the ground truth correction sentence to the 'm2' form. For all the valid editing sets we chose the one which minimized the editing distance.

For the SMT GEC model, we used different data granularities. S_{char} was trained on char-level data and S_{word} was trained on word-level data. Because most characters in Chinese have their own meanings, so it is reasonable to train a char-level SMT GEC model. We simply split every char by space to get the char-level data, and we used the Jieba [28] segmentation tool to split the word in a sentence by space to produce the word-level data.

For the NMT GEC model, we used the pre-trained embedding in different parts of the model. The first choice was to use it for the whole model, which forced the model to learn a proper embedding by itself. Considering the dataset is not large enough for the model to learn the embedding from scratch, we also tested the pre-trained embedding used for both encoder and decoder parts. But the embedding was trained on the Gigaword [10], which was quite different from the sentences written by CFL learners, so we also used the pre-trained embedding only in the decoder part. The configurations of our four different NMT GEC models N_j, $j \in \{1, 2, 3, 4\}$ are shown in Table 1. For the 'Network' column, the 'BiLSTM' means bi-directional LSTM [26].

5.2 Experimental Results

As described in Sect. 4.2, we used our own scorer during the contest, and used the scorer tool released by NLPCC to assess the performance again after the contest. It is worth to mention that the official scorer was released after the contest, so we chose the model combination based on the unofficial scorer written by ourselves. Because the official document released before contest used the F_1 score as the evaluation example, we calculated the F_1 score in our unofficial scorer instead of $F_{0.5}$ score. According to the evaluation of the single model performance of four NMT GEC models by our unofficial scorer, we ranked those models as $N_3 > N_4 > N_1 > N_2$, which determined the order of combination of NMT GEC models in Table 2. In Tables 2 and 3, the 'Precision', 'Recall', and '$F_{0.5}$ (official)' columns are calculated by the official scorer, and the 'F_1 (unofficial)' column is generated by our own scorer.

Table 1. Configurations of four NMT models

Model	Network	Embed	Dataset
N_1	LSTM	No Pre-trained Embedding	NLPCC MinEd
N_2	BiLSTM	Pre-trained Embedding for Encoder and Decoder	NLPCC MinEd
N_3	BiLSTM	Pre-trained Embedding for Encoder and Decoder	NLPCC Expand
N_4	BiLSTM	Pre-trained Embedding Only for Decoder	NLPCC Expand

According to the results shown in Table 2, we chose $S_{char} + S_{word}$ for the SMT GEC model, and $N_3 + N_4$ with M_3 for the NMT GEC model. For a specific sentence s_i, with the candidate C_{iR} generated by the rule-based model, the combination candidates C_{iS} and C_{iN} generated by the low-level combination of SMT and NMT GEC models separately, we used the high-level combination to generate the final result. According to Table 3, we first explored the influence of different M_i on combining three candidates, and chose the best one to add protection mechanism on it. Because we used the unofficial scorer during the contest, we chose M_2 as the conflict solving method and add P_2 protection as our final submission.

Table 2. Low-level Combination

Model	Solve conflict	Precision	Recall	$F_{0.5}$ (official)	F_1 (unofficial)
S_{char}	None	0.2096	0.0758	0.1549	0.1366
S_{word}	None	0.2107	0.0597	0.1399	0.1090
$S_{char} + S_{word}$	M_3	0.2376	0.0928	**0.1811**	**0.1462**
N_3	M_0	0.362	0.0996	0.2371	0.1166
$N_3 + N_4$	M_0	0.3453	0.1196	**0.2507**	**0.1260**
$N_3 + N_4 + N_1 + N_2$	M_0	0.3187	0.1292	0.2464	0.1152
$N_3 + N_4$	M_1	0.3363	0.1283	0.2540	0.1266
$N_3 + N_4$	M_2	0.3433	0.1130	0.2439	0.1259
$N_3 + N_4$	M_3	0.3485	0.1241	0.2559	**0.1318**
$N_3 + N_4$	M_4	0.3493	0.1238	**0.2561**	0.1304

Table 3. High-level combination

Model	Solve conflict	Precision	Recall	$F_{0.5}$ (official)	F_1 (unofficial)
$R + S + N$	M_0	0.3321	0.1714	0.2797	0.2731
$R + S + N$	M_1	0.3145	0.1969	0.2809	0.2342
$R + S + N$	M_2	0.3397	0.1664	0.2811	**0.2786**
$R + S + N$	M_3	0.3382	0.1782	**0.2867**	0.2370
$R + S + N$	M_4	0.336	0.1781	0.2854	0.2573
$R + S + N$	$M_2 + P1$	0.3528	0.1622	0.2856	0.2853
$R + S + N$	$M_2 + P2$	0.4100	0.1375	**0.2936**	**0.3371**

5.3 Case Analysis

We picked up some cases from the test dataset to illustrate the strengths and weaknesses of models in different categories.

As shown in Table 4, different models focus on different types of errors, so it is necessary to combine candidates generated by different models. The rule-based model is good at correcting errors which share similar intonation or font with the original character, such as 工司 to 公司 (similar intonation), 持别 to 特别 (similar font) and so on. The rule-based model can also solve a more complicated situation defined as character-order problem in a word, for example, correct 富 to 富裕 (change the order of characters). The SMT GEC model can also cope with some error-writing characters if they appear frequently in the training corpus. However, the most useful field of this model is to deal with errors which need adding or deleting a character, such as 都一起 to 一起 and so on. The NMT GEC model is good at correcting some complex errors, which need change the whole word with several characters, or reorder and add characters at the same time. For example, it can correct 太污染 to 污染太严重, and 对我看来 to 对我来, and so on, which to some extent 'understand' the collocation of characters.

As shown in Table 5, there still exists many limitations for those GEC models. The rule-based model would accidentally hurt some correct words if the statistical threshold is not set properly, such as 痴想 to 吃香. Although the SMT GEC

Table 4. The cases which can be corrected by our GEC models

Model	Original Sentence	Correction Candidate Generated
Rule	找工作的人比**工**司更多。	找工作的人比**公**司更多。
Rule	现在人们会认为中国，**持**别是北京	现在人们会认为中国，**特**别是北京
Rule	大家希望**裕富**的生活。	大家希望**富裕**的生活。
SMT	官吏腐败，尤其几个贪官污吏老是**枪**夺人民的财产。	官吏腐败，尤其**是**几个贪官污吏老是**枪**夺人民的财产。
SMT	**大使管**所查到的都可以上网。	**大使馆**所查到的都可以上网。
SMT	我和家人**都一起**在关岛过了很幸福的时间。	我和家人**一起**在关岛过了很幸福的时间。
NMT	对我**看来**，曼古空气污染的问题与日俱增。	对我**来说**，曼古空气污染的问题与日俱增。
NMT	北京的空气**太污染**了。	北京的空气**污染太严重**了。
NMT	大学是让学生们能**作为**一名好成人，好的社会的一部分。	大学是让学生们能**成为**一名好成人，好的社会的一部分。
NMT	结果全世界的环境问题**越严重**。	结果全世界的环境问题**越来越严重**。

Table 5. The cases which cannot be corrected by our GEC models

Model	Original Sentence	Correction Candidate Generated
Rule	因此，以自国文化的价值标准来批评别的文化是不合理的，而且应当抛弃的一种**痴想**。	因此，以自国文化的价值标准来批评别的文化是不合理的，而且应当抛弃的一种**吃香**。
SMT	学习是以得到智慧，不是**以**就业。	学习是以得到**的**智慧，不是就业。
NMT	大学的功能不是这样的。**就是学的**。	大学的功能不是这样的。

model can deal with errors about adding or deleting a character, it sometimes would add or delete wrong characters, such as add 的 and delete 以. The NMT GEC model sometimes would direct throw away a part of the sentence if it is too difficult to correct, as the part 就是学的 in the table.

6 Conclusion and Future Work

In this paper, we proposed a system for the GEC task, which combined statistical and neural models. This method consisted of two modules: the correction module and the combination module. In the correction module, two statistical models, including a rule-based model and an SMT GEC model, and an NMT GEC model generated correction candidates for each input sentence. In the combination module, we implemented it in a hierarchical manner. In the low-level combination, we combined models with different configurations within the same category. Then, in the high-level combination, we combined candidates of two statistical models and the neural model generated in the low-level combination. Our system reached the second place on the leaderboard released by the official.

In the future, we will further explore the strengths as well as limitations of three GEC models and combination methods in our system. We will focus on improving the 'Recall' metric of our system.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Brockett, C., Dolan, W.B., Gamon, M.: 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, pp. 249–256. Association for Computational Linguistics (2006)
3. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2), 263–311 (1993)
4. Bustamante, F.R., León, F.S.: GramCheck: a grammar and style checker. In: Proceedings of the 16th Conference on Computational Linguistics-Volume 1, pp. 175–181. Association for Computational Linguistics (1996)
5. Chang, R.Y., Wu, C.H., Prasetyo, P.K.: Error diagnosis of Chinese sentences using inductive learning algorithm and decomposition-based testing mechanism. ACM Trans. Asian Lang. Inf. Process. (TALIP) **11**(1), 3 (2012)
6. Chang, T.H., Sung, Y.T., Hong, J.F., Chang, J.I.: KNGED: a tool for grammatical error diagnosis of Chinese sentences. In: 22nd International Conference on Computers in Education, ICCE 2014. Asia-Pacific Society for Computers in Education (2014)
7. Cheng, S.M., Yu, C.H., Chen, H.H.: Chinese word ordering errors detection and correction for non-native Chinese language learners. In: Proceedings of COLING 2014, The 25th International Conference on Computational Linguistics: Technical Papers, pp. 279–289 (2014)

8. Felice, M., Yuan, Z., Andersen, Ø.E., Yannakoudakis, H., Kochmar, E.: Grammatical error correction using hybrid systems and type filtering. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pp. 15–24 (2014)
9. Gaoqi, R., Zhang, B., Endong, X., Lee, L.H.: IJCNLP-2017 task 1: Chinese grammatical error diagnosis. In: Proceedings of the IJCNLP 2017, Shared Tasks, pp. 1–8 (2017)
10. Gra, D., Chen, K.: Chinese gigaword. LDC Catalog No.: LDC2003T09, ISBN 1, 58563-58230 (2005)
11. Han, N.R., Chodorow, M., Leacock, C.: Detecting errors in English article usage with a maximum entropy classifier trained on a large, diverse corpus. In: LREC (2004)
12. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, pp. 187–197. Association for Computational Linguistics (2011)
13. Heidorn, G.E., Jensen, K., Miller, L.A., Byrd, R.J., Chodorow, M.S.: The EPISTLE text-critiquing system. IBM Syst. J. **21**(3), 305–326 (1982)
14. Ji, J., Wang, Q., Toutanova, K., Gong, Y., Truong, S., Gao, J.: A nested attention neural hybrid model for grammatical error correction. arXiv preprint arXiv:1707.02026 (2017)
15. Junczys-Dowmunt, M., Grundkiewicz, R.: 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, pp. 25–33 (2014)
16. Lee, L.H., Gaoqi, R., Yu, L.C., Endong, X., Zhang, B., Chang, L.P.: Overview of NLP-TEA 2016 shared task for Chinese grammatical error diagnosis. In: Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016), pp. 40–48 (2016)
17. Lee, L.H., Yu, L.C., Chang, L.: Overview of the NLP-TEA 2015 shared task for Chinese grammatical error diagnosis, 07 March 2015
18. Lee, L.H., Yu, L.C., Lee, K.C., Tseng, Y.H., Chang, L.P., Chen, H.H.: A sentence judgment system for grammatical error detection. In: Proceedings of COLING 2014, The 25th International Conference on Computational Linguistics: System Demonstrations, pp. 67–70 (2014)
19. Lin, C.J., Chan, S.H.: Description of NTOU Chinese grammar checker in CFL 2014. In: Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2014), Nara, Japan, pp. 75–78 (2014)
20. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
21. Madnani, N., Tetreault, J., Chodorow, M.: Exploring grammatical error correction with not-so-crummy machine translation. In: Proceedings of the Seventh Workshop on Building Educational Applications using NLP, pp. 44–53. Association for Computational Linguistics (2012)
22. Napoles, C., Callison-Burch, C.: Systematically adapting machine translation for grammatical error correction. In: Proceedings of the 12th Workshop on Innovative use of NLP for Building Educational Applications, pp. 345–356 (2017)
23. Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., Bryant, C.: The CoNLL-2014 shared task on grammatical error correction. In: Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–14 (2014)

24. Ng, H.T., Wu, S.M., Wu, Y., Hadiwinoto, C., Tetreault, J.: The CoNLL-2013 shared task on grammatical error correction (2013)
25. Rozovskaya, A., Roth, D.: Algorithm selection and model adaptation for ESL correction tasks. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 924–933. Association for Computational Linguistics (2011)
26. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
27. Sun, C., Jin, X., Lin, L., Zhao, Y., Wang, X.: Convolutional neural networks for correcting English article errors. In: Li, J., Ji, H., Zhao, D., Feng, Y. (eds.) NLPCC 2015. LNCS (LNAI), vol. 9362, pp. 102–110. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25207-0_9
28. Sun, J.: 'jieba' Chinese word segmentation tool (2012)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
30. Tetreault, J.R., Chodorow, M.: The ups and downs of preposition error detection in ESL writing. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, pp. 865–872. Association for Computational Linguistics (2008)
31. Wu, X., Huang, P., Wang, J., Guo, Q., Xu, Y., Chen, C.: Chinese grammatical error diagnosis system based on hybrid model. In: Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications, pp. 117–125 (2015)
32. Yu, L.C., Lee, L.H., Chang, L.P.: Overview of grammatical error diagnosis for learning Chinese as a foreign language. In: Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications, pp. 42–47 (2014)
33. Yuan, Z., Briscoe, T.: Grammatical error correction using neural machine translation. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 380–386 (2016)
34. Yuan, Z., Felice, M.: Constrained grammatical error correction using statistical machine translation. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pp. 52–61 (2013)
35. Zampieri, M., Tan, L.: Grammatical error detection with limited training data: the case of Chinese. In: Proceedings of ICCE (2014)