Chinese Spelling Check based on Neural Machine Translation

Chiao-Wen Li[1], **Jhih-Jie Chen**[2], **Jason S. Chang**[2]
[1]Institute of Information Systems and Applications
National Tsing Hua University
[2]Department of Computer Science
National Tsing Hua University
{chiaowen, jjc, jason}@nlplab.cc

Abstract

This paper presents a method for Chinese spelling check that automatically learns to correct a sentence with potential spelling errors. In our approach, a character-based neural machine translation (NMT) model is trained to translate a potentially misspelled sentence into correct one, using right-and-wrong sentence pairs from newspaper edit logs and artificially generated data. The method involves extracting sentences containing edits of spelling correction from edit logs, using commonly confused right-and-wrong word pairs to generate artificial right-and-wrong sentence pairs in order to expand our training data, and training the NMT model. The evaluation on the United Daily News (UDN) Edit Logs and SIGHAN-7 Shared Task shows that adding artificial error data significantly improves the performance of Chinese spelling check system.

1 Introduction

Spelling check is a common yet important task in natural language processing. It plays an important role in a wide range of applications such as word processors, assisted writing systems, and search engines. For example, Web search engines such as *Google* and *Bing* typically perform spelling check on queries, in order to retrieve documents better meeting the user's information need. In contrast to Web search engines, while *Microsoft Word* has a very effective spelling checker for English, there is still considerable room to improve one for Chinese. Compared to Western languages (e.g., English and German), relatively little work has been done on Chinese spelling check because there are more challenges such as unclear word boundaries and massive characters in Chinese. Moreover, lack of training data hinders the development of Chinese spelling check.

Consider the sentence "他在文學方面有很高的造旨。", in which the character "旨" is a typo. For another sentence "他在文學方面有很高的造藝。", the character "藝" is also a typo. The correct character of the two typos should be "詣". Chinese spelling errors often stem from two main reasons: one is similar sound (e.g., *藝 and 詣) and the other is similar shape (e.g., *旨 and 詣), as pointed by Liu et al. (2011).

Intuitively, a spelling error can be automatically corrected more precisely with machine learning models trained on more data. Unfortunately, there is only limited training data of spelling correction in Chinese, and thus it is no easy to train an NMT model achieving good performance of Chinese spelling check. Therefore, we want to improve the spelling check model by generating artificial errors to increase training data.

In this paper, we present a new system, *AccuSpell*, that automatically learns to generate the corrected sentence for a potentially misspelled sentence using neural machine translation (NMT) model. An example of *AccuSpell* checking for the sentence "在我們的生命中，常會碰到一些措折和失敗。" is shown in Figure 1. *AccuSpell* learns how to effectively correct a given sentence by training on more data, including real edit logs and artificially generated data. We will describe how to generate artificial data and the training process in detail in Section 3.

Figure 1: A screenshot of the system *AccuSpell*

The rest of the article is organized as follows. We review the related work in the next section. Then we describe how to extract misspelled sentences from edit logs and how to generate artificial sentences with typos in Section 3. We also present our method for automatically learning to correct typos in a given sentence. Section 4 describes resources and datasets we used in the experiment. In our evaluation, over two test sets, we compare the performance of several models in Section 5. Finally, we summarize and point out the future work in Section 6.

2 Related Work

Error Correction has been an area of active research, which involves Grammatical Error Correction (GEC) and Spelling Error Correction (SEC). Recently, researchers have begun applying neural machine translation models to both GEC and SEC, and gained significant improvement (e.g., Yuan and Briscoe (2016) and Xie et al. (2016)). However, compared to English, relatively little work has been done on Chinese error correction. In our work, we address the Chinese spelling error correction task on text written by native speakers, and an improved model by generating artificial typos.

Early works on Chinese spelling check typically focused on rule-based and statistical approaches. Rule-based approaches usually use dictionary to identify typos and confusion set to find possible corrections, while statistical methods use the noisy channel model to find candidates of correction for a typo, and language model to calculate the likelihood of the corrected sentences. Chang (1995) proposed an approach that integrates both rule-based method and statistical method to automatically correct Chinese spelling errors. The approach involves a confusing character substitution mechanism and bigram language model. Later, Zhang et al. (2000) pointed out that the method proposed by Chang (1995) only address character substitution errors, other kinds of errors such as deletion and insertion can not be handled. They proposed a similar approach using confusing word substitution and trigram language model to extend the method proposed by Chang (1995).

In recent years, Statistical Machine Translation (SMT) has been applied to Chinese spelling check. Wu et al. (2010) presented a system using a new error model and a common error template generation method to detect and correct Chinese character errors, which reduce the false alarm rate significantly. The idea of the error model is adopted from the noisy channel model, a framework of SMT, which is used in many NLP tasks such as spelling check and machine translation. Chiu et al. (2013) proposed a data-driven method that detects and corrects Chinese errors based on phrasal statistical machine translation framework. They used word segmentation and dictionary to detect possible spelling errors, and correct the errors using SMT model built from a large corpus.

More recently, Neural Machine Translation (NMT) has been adopted in error correction task and achieved state-of-the-art performance. Yuan and Briscoe (2016) presented the very first NMT model for grammatical error correction of English. However, word-based NMT models usually suffer from rare word problem and infrequent words are substituted for a "UNK" token. Then, a character-based NMT approach was proposed by Xie et al. (2016) to avoid the problem of out-of-vocabulary words. Subsequently, Chollampatt and Ng (2018) proposed a multilayer convolutional encoder-decoder neural network to correct grammatical, orthographic, and collocation errors. Until now, most work on error correction using NMT model aimed at correction for English text. In contrast, we focus on correcting Chinese spelling errors.

Building an error correction system using machine learning techniques typically requires a considerable amount of error-annotated data. Unfortunately, limited availability of error-annotated data is holding back progress in the area of automatic er-

ror correction. Felice and Yuan (2014) presented a method of generating artificial errors for training, and improved NMT models for correcting mistakes made by English as a second language. Rei et al. (2017) investigated two alternative approaches for artificially generating all types of writing errors. They extracted error patterns from an annotated corpus and transplanting them into error-free text. In addition, they built a phrase-based SMT error generator to translate the grammatically correct text into incorrect one.

In a study closer to our work, Gu and Lang (2017) applied sequence-to-sequence (seq2seq) model to construct a word-based Chinese spelling error corrector. They established their own error corpus for training and evaluation by transplanting errors into an error-free news corpus. Comparing with traditional methods, their model can correct errors more effectively.

In contrast to the previous research in Chinese spelling check, we present a system that uses newspaper edit logs to train an NMT model for correcting typos in Chinese text. We also propose a method to generate artificial error data to enhance the NMT model. Additionally, to avoid rare word problem, our NMT model is trained at character level. The experiment results show that our model achieves significantly better performance, especially at a extremely low false alarm rate.

3 Method

We focus on correcting Chinese spelling errors in a given sentence by formulating the spelling check as a machine translation problem. A sentence with typos is treated as the source sentence, which is translated into a target sentence with errors corrected. Thus, we train a neural machine translation (NMT) model on right-and-wrong sentence pairs extracted from newspaper edit logs. Unfortunately, the sentence pairs from newspaper edit logs are too small to train a good NMT model. To develop a more effective Chinese spelling check system, a promising approach is to automatically generate errors in presumably correct sentences for expanding the training data (Felice, 2016), leading the system to cope with a wider variety of errors and contexts.

In our approach, we first extract the sentences with spelling errors from edit logs (Section 3.1) and generate artificial misspelled sentences from a set of error-free sentences (Section 3.2). We then use these data to train the NMT model (Section 3.3).

【記者葉子菁／台北報導】12月台指期合約將於明日結算，台指期今日開高後震盪走低，並回測9<FONT class=1 title=李定強新增, color=#265e8a>,</FONT>200點位置。永豐期貨副總廖祿民表示，台股目前屬於盤跌、慢慢走弱的盤勢，從外資在期貨淨多單的留倉來看，仍未有企圖撐在高點結算的意味，且適逢耶誕假期，外資也不急著<FONT style="TEXT-DECORATION: line-through" class=3 title=李定強刪除, color=#555588>佈</FONT><FONT class=1 title=李定強新增, color=#265e8a>布</FONT>局明年，惟從選擇權的Put/Call Ratio來看，1.4仍屬於多方架構，後續9<FONT class=1 title=李定強新增, color=#265e8a>,</FONT>200點為觀察支撐點位的基礎，而預期在12/5日的低<FONT class=1 title=李定強新增, color=#265e8a>點</FONT>9<FONT class=1 title=李定強新增, color=#265e8a>,</FONT>138點具有較強勁的支撐力道。</P>

Figure 2: An example of edit logs in HTML format

3.1 Extracting Misspelled Sentences from Edit Logs

In this stage, we extract a set of sentences with spelling errors annotated by simple edit tags (i.e., "[- -]" for deletion and "{+ +}" for insertion) from edit logs. For example, the sentence "希望未來主要島嶼都有完善的[-馬-]{+碼+}頭，" contains the edit tags "[-馬-]{+碼+}" that means the original character "馬"was replaced with "碼".

The input to this stage are a set of edit logs in HTML format, containing the name of editor, the action of edit (1 is insertion and 3 is deletion), the target content and some CSS attributes, as shown in Figure 2. We first convert HTML files to simple text files by removing HTML tags and using simple edit tags "{+ +}" and "[- -]" to represent the edit actions of insertion and deletion respectively. For example, the sentence in HTML format

> "外 資 也 不 急 著<FONT style="TEXT-DECORATION: line-through" class=3 title=XXX刪除, color=#555588>佈</FONT><FONT class=1 title=XXX新 增, color=#265e8a>布</FONT>局明年，"

is converted to

> "外 資 也 不 急 著[-佈-]{+布+}局 明 年，".

After that, we attempt to extract the sentences that contain at least one typo. The edit logs could con-

32nd Pacific Asia Conference on Language, Information and Computation
Hong Kong, 1-3 December 2018

- 一些較落後地區（如孟加拉）因表面水體受到[-污-]{+汙+}染，
- 到大陸創業條件首要是膽[-試-]{+識+}，
- [-盡-]{+敬+}請鎖定相關報導。
- 現場{+儼+}[-嚴-]然成為超跑展示中心，
- 十六支球隊要{+爭+}[-整-]取十二張高雄複賽門票。
- 也連續兩年創下歷史新高{+記+}[-紀-]錄。
- 一口氣追回昨天創下英國脫歐以來最大[-鵚-]{+頹+}勢，
- 把施工圍籬變成美麗的彩繪或塗[-鴨-]{+鴉+}，
- 也通報捕狗隊來協助；對受害學童[-己-]{+已+}派員慰問。
- 不論在市[-佔-]{+占+}率、獲利表現、品牌及服務等各方面，

Figure 3: Example outputs for the stage of extracting misspelled sentences

tain many kinds of edits, including spelling correction, content changes, and style modification (such as synonyms replacement). Among these edits, we are only concerned with spelling correction. However, lack of edit type annotation makes it difficult to directly identify spelling correction. Thus, we consider consecutive single-character edit pairs of deletion and insertion (e.g., "[-佈-]{+布+}" or "{+布+}[-佈-]") as spelling correction, and extract the sentences containing such edit pairs. Finally, we obtain a set of sentences with spelling errors annotated using simple edit tags, as shown in Figure 3.

3.2 Generating Artificially Misspelled Sentences

To make our Chinese spelling check system more effective, we create a set of artificial misspelled sentences for expanding our training data.

The input to this stage are a set of presumably error-free sentences from news articles with word segmentation done using a word segmentation tool provided by the CKIP Project (Ma and Chen, 2003). Artificially misspelled sentences are generated by injecting errors into these error-free sentences. Although a correct word could be misspelled as any other Chinese word, some right-and-wrong word pairs are more likely to happen than others. In order to generate realistic spelling errors, we use a confusion set consisting of commonly confused right-and-wrong word pairs (see Table 1). The wrong words in confusion set are used to replace counterpart correct words in the sentences. For example, we use error-free sentence "也 跟 患者 賠罪 了 十 分鐘" to generate three misspelled sentences, as shown in Table 2. The output of this stage is a set of right-and-wrong sentence pairs.

Table 1: Examples of confusion set

Correct Word	Wrong Words
部署	布署, 部處, 佈署, 步署
賠罪	培罪, 陪罪

Table 2: Artificial misspelled sentences for "也 跟 患者 賠罪 了 十 分鐘"

Artificial Misspelled Sentence	Wrong Word
也 跟 患者 **培罪** 了 十 分鐘 ；	培罪
也 跟 患者 **陪罪** 了 十 分鐘 ；	陪罪
也 跟 患者 賠罪 了 十 **分鍾** ；	分鍾

The confusion set plays an important role in this stage, so it is critical to decide what kinds of confusion set to use. There are several available word-level and character-level confusion sets. However, compare to word-level, a Chinese character could be confused with more other characters based on shape and sound similarity. For example, the character "賠" is confused with 23 characters with similar shape and 21 characters with similar sound in a character-level confusion set, while the word "賠罪" is confused with only two words in a word-level confusion set. Moreover, an occurring typo might involve not only the character itself but also the context. If we use the character-level confusion set, an error-free sentence would produce numerous and probably unrealistic artificial misspelled sentences. Therefore, we decide to use word-level confusion sets.

3.3 Neural Machine Translation Model

We train a character-based neural machine translation (NMT) model for developing a Chinese spelling checker, which translates a potentially misspelled sentence into a correct one.

The architecture of NMT model typically consists of an encoder and a decoder. The encoder consumes the source sentence $X = [x_1, x_2, \ldots, x_I]$ and the decoder generates translated target sentence $Y = [y_1, y_2, \ldots, y_J]$. For the task of correcting spelling errors, a potentially misspelled sentence is treated as the source sentence X, which is translated into the target sentence Y with errors cor-

rected. To train the NMT model, we use a set of right-and-wrong sentence pairs from edit logs (Section 3.1) and artificially-generated data (Section 3.2) as target-and-source training sentence pairs.

In the training phase, the model is given (X, Y) pairs. At encoding time, the encoder reads and transforms a source sentence X, which is projected to a sequence of embedding vectors $\mathbf{e} = [e_1, e_2, \ldots, e_I]$, into a context vector $\mathbf{c}$:

$$c = q(h_1, h_2, ..., h_I) \tag{1}$$

where q is some nonlinear function.

We use a bidirectional recurrent neural network (RNN) encoder to compute a sequence of hidden state vectors $\mathbf{h} = [h_1, h_2, ..., h_I]$. The bidirectional RNN encoder consists of two independent encoders: a forward and a backward RNN. The forward RNN encodes the normal sequence, and the backward RNN encodes the reversed sequence. A hidden state vector h_i at time i is defined as:

$$fh_i = \text{ForwardRNN}(h_{i-1}, e_i) \tag{2}$$
$$bh_i = \text{BackwardRNN}(h_{i+1}, e_i) \tag{3}$$
$$h_i = [fh_i || bh_i] \tag{4}$$

where $||$ denotes the vector concatenation operator.

At decoding time, the decoder is trained to output a target sentence Y by predicting the next character y_j based on the context vector c and all the previously predicted characters $\{y_1, y_2, \ldots, y_{j-1}\}$:

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{j=1}^{J} p(y_j|y_1, y_2, ..., y_{j-1}; c) \tag{5}$$

The conditional probability is modeled as:

$$p(y_j|y_1, y_2, ..., y_{j-1}; c) = g(y_{j-1}, h'_j, c) \tag{6}$$

where g is a nonlinear function, and h'_j is the hidden state vector of the RNN decoder at time j.

We use a attention-based RNN decoder that focuses on the most relevant information in the source sentence rather than the entire source sentence. Thus, the conditional probability in Equation 5 is redefined as:

$$p(y_j|y_1, y_2, ..., y_{j-1}; \mathbf{e}) = g(y_{j-1}, h'_j, \mathbf{c_j}) \tag{7}$$

where the hidden state vector h'_j is computed as follow:

$$h'_j = f(y_{j-1}, h'_{j-1}, \mathbf{c_j}) \tag{8}$$
$$c_j = \sum_{i=1}^{I} a_{ji} h_i \tag{9}$$
$$a_{ji} = \frac{\exp(\text{score}(h'_j, h_i))}{\sum_{i'=1}^{I} \exp(\text{score}(h'_j, h_{i'}))} \tag{10}$$

Unlike Equation 6, here the probability is conditioned on a different context vector c_j for each target character y_j. The context vector c_j follows the same computation as in Bahdanau et al. (2014). We use the global attention approach with general score function to compute the attention weight a_{ji}:

$$\text{score}(h'_j, h_i) = {h'_j}^{\mathrm{T}} W_a h_i \tag{11}$$

Instead of implementing an NMT model from scratch, we use *OpenNMT* (Klein et al., 2017), an open source toolkit for neural machine translation and sequence modeling, to train the model. The training details and hyper-parameters of our model will be described in Section 4.2

4 Experimental Setting

In this section, we first give a brief description of the datasets used in the experiments in Section 4.1, and describe the hyper-parameters for the NMT model in Section 4.2. Then several NMT models with different experimental setting for comparing performance are described in Section 4.3. Finally in Section 4.4, we introduce the evaluation metrics for evaluating the performance of these models.

4.1 Dataset

United Daily News (UDN) Edit Logs: UDN Edit Logs was provided to us by UDN Digital. This dataset records the editing actions of daily UDN news from June 2016 to January 2017. There are 1.07 million HTML files with more than 30 million edits of various types, with approximately 11 million insertions and 20 million deletions. We extracted a set of annotated sentences involving spelling error correction from this edit logs using the approach described in Section 3.1. To train on NMT model, we transformed every annotated sentence into a source-and-target parallel sentence. For example, "外資

Table 3: Number of word pairs of five confusion sets

Confusion Set	Number of confused word pairs
聯合報統一用字	1,056
東東錯別字	38,125
新編常用錯別字門診	492
常見錯別字辨正辭典	601
國中錯字表	1,460

也不急著[-佈-]{+布+}局明年，” is transformed into a source sentence “外資也不急著佈局明年，” and a target sentence “外資也不急著布局明年，”. In total, there are 238,585 sentences extracted from UDN Edit Logs, and each sentence contains only edits related to spelling errors. We divided these extracted sentences into two parts: one (226,913 sentences) for training NMT models, and the other (11,943 sentences) for evaluation in our experiments.

United Daily News (UDN): The UDN news dataset was also provided by UDN Digital. This dataset consists of published newswire data from 2004 to 2017, which contains approximately 1.8 million news articles with over 530 million words. Unlike UDN Edit Logs, UDN are composed of news articles which had been edited and published. We used the presumably error-free sentences in this dataset to generate artificially misspelled sentences, as described in Section 3.2.

Confusion Set: We collect five confusion sets from online and print publications: 聯合報統一用字, 東東錯別字, 新編常用錯別字門診(蔡有秩, 2003), 常見錯別字辨正辭典(蔡榮圳, 2012), and 國中錯字表. The confused word pairs of five confusion sets (see Table 3) are combined into a collection with over 40,000 word pairs. However, for a given confused word pair, the judgments in different confusion sets might be inconsistent. Consider a confused word pair [“鐘錶”, “鐘表”]. “鐘錶” is right and “鐘表” is wrong in 東東錯別字, while “鐘表” is adopted and “鐘錶” is not recommended in 聯合報統一用字. Furthermore, the confusion sets are not guaranteed to be absolutely correct. To resolve these problems, we used the Chinese dictionary published by Ministry of Education of Taiwan as the gold standard. After filtering out the invalid word pairs, 33,551 distinct commonly confused word pairs were obtained.

Test Data: We used two test sets for evaluation:

- **UDN Edit Logs**: As mentioned earlier, UDN Edit Logs were partitioned into two independent parts, for training and testing respectively. The test part contains 11,943 sentences and we only used 1,175 sentences for evaluation, 919 out of which contain at least one error.

- **SIGHAN-7**: We also used the dataset provided by SIGHAN 7 Bake-off 2013 ((Wu et al., 2013)). This dataset contains two subtasks: Subtask 1 is for error detection and Subtask 2 is for error correction. In our work, we focus on evaluating error correction, so we used Subtask 2 as an additional test set. There are 1,000 sentences with spelling errors in Subtask 2, and the average length of sentences is approximately 70 characters. To evaluate the false alarm rate of our system, we segmented these sentences into 6,101 clauses, and 1,222 of which contain at least one error, and the remainder are error-free.

4.2 Hyper-parameters of NMT Model

We trained several models using the same hyper-parameters in our experiments. For all models, the source and target vocabulary sizes are limited to 10K since the models are trained at character level. For source and target characters, the character embedding vector size is set to 500. We trained the models with sequences length up to 50 characters for both source and target sentences.

The encoder is a 2-layer bidirectional long-short term memory (LSTM) networks, which consists of a forward LSTM and a backward LSTM, and the decoder is also a 2-layer LSTM. Both the encoder and the decoder have 500 hidden units. We use the Adam Algorithm (Kingma and Ba, 2014) as the optimization method to train our models with learning rate 0.001, and the maximum gradient norm is set to 5. Once a model is trained, beam search with beam size set to 5 is used to find a translation that approximately maximizes the probability.

Table 4: The number of training sentences of the 7 models

Model	UDN Edit Logs	Artificially Generated Data
UDN-only	226,913	-
UDN+ART (1:1)	226,913	225,985
UDN+ART (1:2)	226,913	440,143
UDN+ART (1:3)	226,913	673,006
UDN+ART (1:4)	226,913	899,385
ART-only	-	899,385
FEAT-Sound&Shape	226,913	673,006

* *FEAT-Sound&Shape* is trained on the same data in *UDN+ART (1:3)*

4.3 Models Compared

We use the training part of UDN Edit Logs and the artificially generated misspelled sentences as the training data. To investigate whether the artificially generated data improves the performance of our Chinese spelling check model, we compared the results produced by models trained on different combinations of UDN Edit Logs and artificially generated data. In addition, we use the pronunciation and shape of a character as additional features for both the source and target sides to train another model. For example, for the character "詣", the pronunciation feature is "一" (without considering the tone) and the shape features are "言" and "旨".

There are totally seven models trained for comparing, and only last one was trained with features, as shown in Table 4.

4.4 Evaluation Metrics

We use the metrics provided by SIGHAN-8 Bake-off 2015 for Chinese spelling check shared task (Tseng et al., 2015), which include False Positive Rate (FPR), Accuracy, Precision, Recall, and F1, to evaluate our systems.

5 Results and Discussion

Table 5 shows the evaluation results of the two test sets we used. For UDN Edit Logs test set, as we can see, all models trained on edit logs plus artificially generated data perform better than the one trained on only edit logs. Moreover, *UDN-only* performs slightly worse, while *ART-only* performs the worst on all metrics. Though the model trained with sound and shape features has a relatively bad FPR, it has the best performance on accuracy, precision, recall, and F1 score. For the other test set, SIGHAN-7, *UDN+ART (1:4)* performs substantially better than the other models, noticeably improving on all metrics. Interestingly, in contrast to the results of UDN Edit Logs, the model trained on only edit logs has the worst performance, while the model trained on only artificially generated data performs reasonably well. We note that there is no obvious improvement in the performance of the model trained with sound and shape features except the recall.

In general, our systems obtain lower average FPRs on the two test sets. There are two phenomena worth mentioning. First, the model trained on only edit logs (*UDN-only*) performs well on UDN Edit Logs but very poorly on SIGHAN-7. In contrast, the model trained on only artificially generated data (*ART-only*) has worst performance on UDN Edit Logs but acceptable performance on SIGHAN-7 Second, it is worth noting that the model trained with sound and shape features has significantly better accuracy, recall, and F1 score on UDN Edit Logs. However, on SIGHAN-7, only the recall is a little better than the model trained without using features.

Besides the test data, we also found that the model trained with additional features could correct some new and unseen errors. For example, the sentence "他在文學方面有很高的造酯。" with a typo "酯", which is not corrected by a model trained without features probably because of the non-existence in the training data. However, the sentence is translated correctly into "他在文學方面有很高的造詣。" by the model trained with additional sound and shape features.

Moreover, to prove that NMT-based method performs better than traditional methods, we compare the evaluation results of our NMT models with dictionary-based models, as shown in Table 6. The **UDN** dictionary contains a set of right-and-wrong word pairs from the training part of UDN Edit Logs, and the **CONF** dictionary is the confusion set we used to generate artificial error data. We use the dictionaries to correct errors directly. Specifically, we search errors in text and replace them with counter-

Table 5: Evaluation on UDN Edit Logs and SIGHAN-7 test set

Test Set	Model	FPR	Accuracy	Precision	Recall	F1
UDN Edit Logs	UDN-only	.066	.64	.80	.64	.71
	UDN+ART (1:1)	.090	.69	.84	.69	.76
	UDN+ART (1:2)	.063	.71	.86	**.72**	.78
	UDN+ART (1:3)	.066	.70	.86	.69	.76
	UDN+ART (1:4)	**.059**	.71	.87	.71	.78
	ART-only	.137	.35	.43	.26	.33
	FEAT-Sound&Shape	.098	**.72**	**.88**	**.72**	**.79**
SIGHAN-7	UDN-only	.109	.74	.19	.17	.18
	UDN+ART (1:1)	.089	.83	.50	.59	.54
	UDN+ART (1:2)	.081	.84	.54	.61	.57
	UDN+ART (1:3)	.078	**.85**	.56	.62	.58
	UDN+ART (1:4)	**.073**	**.85**	**.58**	.63	**.61**
	ART-only	.079	.84	.53	.58	.56
	FEAT-Sound&Shape	.097	.83	.51	**.64**	.57

Table 6: Evaluation results (recall rate) of dictionary-based approach

Model	UDN Edit Logs	SIGHAN-7
Dictionary-based$_{UDN}$	.13	.04
UDN-only	.64	.17
Dictionary-based$_{CONF}$	.19	.49
ART-only	.26	.58
Dictionary-based$_{UDN+CONF}$	.19	.38
UDN+ART (1:4)	.71	.63

parts in the dictionaries. As we can see, the NMT-based models have higher recall than the dictionary-based models.

6 Conclusion

In summary, we have proposed a novel method for learning to correct typos in Chinese text. The method involves combining real edit logs and artificially generated errors to train a NMT model that translates a potentially erroneous sentence into correct one. The results prove that adding artificially generated data successfully improves the overall performance of error correction. We also found that some unseen errors might be corrected using NMT model.

Many avenues exist for future research and improvement of our system. For example, the method for extracting misspelled sentences from newspaper edit logs could be improved. When extracting, we only consider the sentences contain consecutive single-character edit pairs. However, two-character edit pairs could also involve spelling correction. Moreover, we could investigate how to use character-level confusion sets to expand the scale of confused word pairs. If we have more possibly confused word pairs, we could generate more comprehensive artificial error data. Additionally, an interesting direction to explore is expanding the scope of error correction to include grammatical errors. Yet another direction of research would be to consider focusing on implementing the neural machine translation model for Chinese spelling check. In our work, we pay more attention to the aspect of data, so relatively less experiments were done for tuning parameters of model.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chao-Huang Chang. 1995. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, volume 95, pages 278–283. Citeseer.

Hsun-wen Chiu, Jian-cheng Wu, and Jason S Chang. 2013. Chinese spelling checker based on statistical machine translation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 49–53.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. *arXiv preprint arXiv:1801.08831*.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the EACL*, pages 116–126.

Mariano Felice. 2016. Artificial error generation for translation-based grammatical error correction. Technical report, University of Cambridge, Computer Laboratory.

Sunyan Gu and Fei Lang. 2017. A chinese text corrector based on seq2seq model. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2017 International Conference on*, pages 322–325. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

C-L Liu, M-H Lai, K-W Tien, Y-H Chuang, S-H Wu, and C-Y Lee. 2011. Visually and phonologically similar characters in incorrect chinese words: Analyses, identification, and applications. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):10.

Wei-Yun Ma and Keh-Jiann Chen. 2003. Introduction to ckip chinese word segmentation system for the first international chinese word segmentation bakeoff. In *Proceedings of the 2nd SIGHAN on CLP*, pages 168–171.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the 8th SIGHAN Workshop on CLP*, pages 32–37.

Shih-Hung Wu, Yong-Zhi Chen, Ping-Che Yang, Tsun Ku, and Chao-Lin Liu. 2010. Reducing the false alarm rate of chinese character error detection and correction. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42.

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 2016 NAACL-HLT*, pages 380–386.

Lei Zhang, Changning Huang, Ming Zhou, and Haihua Pan. 2000. Automatic detecting/correcting errors in chinese text by an approximate word-matching algorithm. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 248–254. Association for Computational Linguistics.

蔡有秩. 2003. 新編錯別字門診. 語文訓練叢書. 螢火蟲.

蔡榮圳. 2012. 常見錯別字辨正辭典. 中文可以更好. 商周出版.