Computational Linguistics and Chinese Language Processing
Vol. 25, No. 1, June 2020, pp. 1-28

Chinese Spelling Check based on Neural Machine Translation

Jhih-Jie Chen*, Hai-Lun Tu+, Ching-Yu Yang*,

Chiao-Wen Li# and Jason S. Chang*

Abstract

We present 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 the 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 contain edit 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 can significantly improve the performance of Chinese spelling check system.

Keywords: Chinese Spelling Check, Artificial Error Generation, Neural Machine Translation, Edit Log

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, search engine without spelling check is not user-friendly, while assisted writing system must perform spelling check as the minimal requirement. Web search engines such as Google (www.google.com) and Bing

* Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan
+Department of Library and Information Science, Research and Development Center for Physical Education, Health, and Information Technology, Fu Jen Catholic University, New Taipei, Taiwan
Department of Information System and Application, National Tsing-Hua University, Hsinchu, Taiwan
E-mail: {jjc, helentu, chingyu, chiaowen, jason}@nlplab.cc

(www.bing.com) typically perform spelling check on queries, in order to retrieve documents better meeting the user information need. The users' queries would be corrected first by the spelling check component in order to avoid irrelevant or low-quality search results. In contrast to Web search engines, while Microsoft Word has a very effective spelling checker for English, there is still considerable room to improve the one for Chinese.

Consider a sentence "他在文學方面有很高的造旨。" ('He is highly accomplished in literature.'). In the context of this sentence, the character "旨" (pronounced 'zhi') is a typo. For another sentence "他在文學方面有很高的造藝。", the character "藝" (pronounced 'yi') is also a typo. For these two typos, the correct character is "詣" (pronounced 'yi'). Chinese spelling errors are due to 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).

Unfortunately, such spelling error is probably uneasy to correct due to limited training data. In fact, there is a lack of training data for the Chinese spelling check task. Compared to western languages (e.g., English and German), relatively little work has been done on Chinese spelling check and few datasets are available. More spelling errors can be corrected with a machine learning model trained on more data. It could be that there are some fundamental problems such as no word boundaries, too many characters, and inconsistent use along time. Chinese spelling check could be more practical if more training data is available.

One solution to the lack of training data is to create artificial one for training. Researches on artificial error generation for English have shown great potential in improving underlying models for writing error correction (Felice & Yuan, 2014; Rei, Felice, Yuan, & Briscoe, 2017). In other words, by generating artificial errors to increase data, we might have a chance to make spelling check models better and stronger. However, very few works have focused on generating artificial errors for Chinese.

In this paper, we present *AccuSpell*, a system that automatically learns to generate the corrected sentence for a potentially misspelled sentence using neural machine translation (NMT) model. The system is built on a new dataset consisting of edit logs of journalists from the United Daily News (UDN). Moreover, we collect a number of confusion set for generating artificial errors to augment the data for training. The evaluation on the UDN Edit Logs and SIGHAN-7 Shared Task shows that adding artificial error data can significantly improve the performance of Chinese spelling check system. The model is deployed on Web and an example *AccuSpell* searches for the sentence "今晚月色很美，我想小灼一杯。" ('The moon is so beautiful tonight, and I want a drink.') is shown in Figure 1. *AccuSpell* has determined that "今晚月色很美，我想小酌一杯。" is the most probably corrected sentence. *AccuSpell* learns how to effectively correct a given sentence during training by using more data, including real edit logs and artificially generated data. We will describe how to

create artificial data and training process in detail in Section 3.

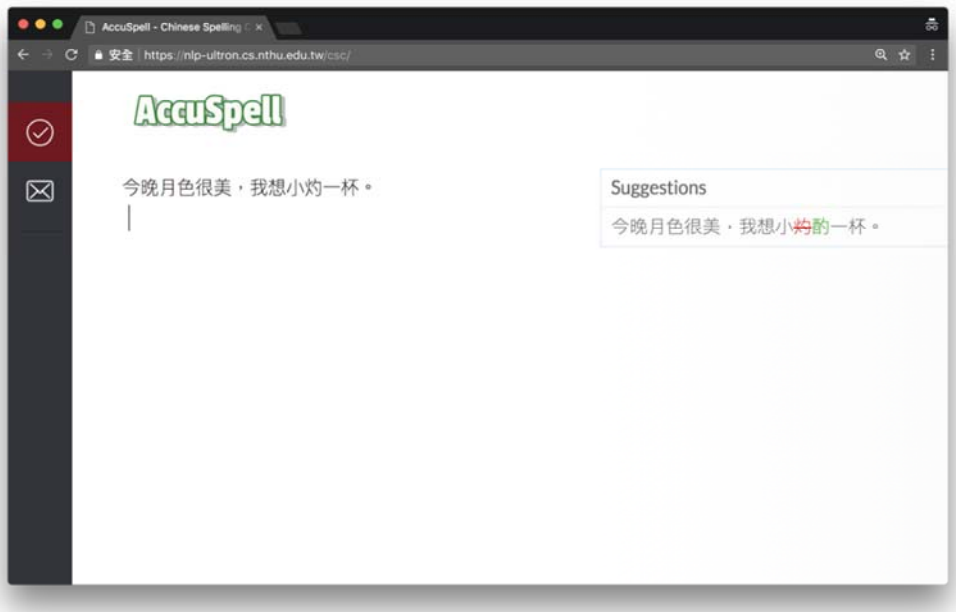

Figure 1. An example the Web version of AccuSpell searches for input "今晚月色很美，我想小灼一杯。" ('The moon is so beautiful tonight, and I want a drink.')

At run-time, *AccuSpell* starts with a sentence or paragraph submitted by the user (e.g., "今晚月色很美，我想小灼一杯。"), which was first divided into clauses. Each clause then is splitted into Chinese characters before being fed to the NMT model. Finally, the model outputs an n-best list of sentences. In our prototype, *AccuSpell* returns the best sentence to the user directly (see Figure 1); alternatively, the best sentence returned by *AccuSpell* can be passed on to other applications such as automatic essay rater and assisted writing systems.

The rest of the article is organized as follows. We review the related work in the next section. Then we describe how to extract the misspelled sentences from newspaper 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 the resources and datasets we used in the experiment. In our evaluation, over two set of test data, we compare the performance of several models trained on both real and artificial data with the model trained on only real data 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 & Briscoe, 2016; Xie, Avati, Arivazhagan, Jurafsky, & Ng, 2016). However, compared to English, relatively little work has been done on Chinese error correction. In our work, we address the spelling error correction task, that focuses on generating corrections related to typos in Chinese text written by native speakers.

Early work on Chinese spelling check typically uses 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 combines rule-based method and statistical method to automatically correct Chinese spelling errors. The approach involves confusing character substitution mechanism and bigram language model. They used a confusion set to replace each character in the given sentence with its corresponding confusing characters one by one, and use a bigram language model built from a newspaper corpus to score all modified sentences in an attempt to find the best corrected sentence. Zhang, Huang, Zhou, and Pan (2000) pointed out that Chang (1995)'s method can only address character substitution errors, other kinds of errors such as character deletion and insertion cannot be handled. They proposed an 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, Chen, Yang, Ku and Liu (2010) presented a system using a new error model and a common error template generation method to detect and correct Chinese character errors that can reduce false alarm rate significantly. The idea of 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, Wu and Chang (2013) proposed a data-driven method that detect and correct Chinese errors based on phrasal statistical machine translation framework. They used word segmentation and dictionary to detect possible spelling errors, and correct the errors by using SMT model built from a large corpus.

More recently, Neural Machine Translation (NMT) has been adopted in error correction task and has achieved state-of-the-art performance. Yuan and Briscoe (2016) presented the very first NMT model for grammatical error correction of English sentences and proposed a two-step approach to handle the rare word problem in NMT. The word-based NMT models usually suffer from rare word problem. Thus, a neural network-based approach using character-based model for language correction was proposed by Xie *et al*. (2016) to avoid the problem of out-of-vocabulary words. 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 done by using NMT model aimed

at grammatical errors for English text. In contrast, we focus on correcting Chinese spelling errors.

Building an error correction system using machine learning techniques typically require a considerable amount of error-annotated data. Unfortunately, limited availability of error-annotated data is holding back progress in the area of automatic error correction. Felice and Yuan (2014) presented a method that generates artificial errors for correcting grammatical mistakes made by learners of English as a second language. They are the first to use linguistic information such as part-of-speech to refine the contexts of occurring errors and replicate them in native error-free text, but also restricting the method to five error types. 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 an extremely low false alarm rate.

3. Methodology

Submitting a misspelled sentence (e.g., “今晚月色很美，我想小灼一杯。”) to a spelling check system with limited training data often does not work very well. Spelling check systems typically are trained on data of limited size and scope. Unfortunately, it is difficult to obtain a sufficiently large training set that cover most common errors, corrections, and contexts. When encountering new and unseen errors and contexts, these systems might not be able to correct such errors. To develop a more effective spelling check system, a promising approach is to automatically generate artificial errors in presumably correct sentences for expanding the training data, leading the system to cope with a wider variety of errors and contexts.

3.1 Problem Statement

We focus on correcting spelling errors in a given sentence by formulating the Chinese 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. The plausible target sentence predicted by a neural machine translation model is then returned as the output of the system. The returned sentence can be viewed by the users directly as suggestion for correcting a misspelled sentence, or passed on to other applications such as automatic essay rater and assisted writing systems. Thus, it is important that the misspelled characters in a given sentence be corrected as many as possible. At the same time, the system should avoid making false corrections. Therefore, our purpose is to return a sentence with most spelling errors corrected, while keeping false alarms reasonably low. We now formally state the problem that we are addressing.

***Problem Statement*:** We are given a possibly misspelled sentence X with n characters $x_1,x_2,...,x_n$. Our goal is to return the correctly spelled sentence Y with m characters $y_1,y_2,...,y_m$. For this, we prepare a dataset of right-and-wrong sentence pairs in order to train a neural machine translation (NMT) model. The sentences come from real edit logs and artificially-generated data.

In the rest of this section, we describe our solution to this problem. First, we describe the process of automatically learning to correct misspelled sentences in Section 3.2. More specifically, we describe the preprocessing of edit logs in Section 3.2.1, and how to artificially generate similar sentences with edits in Section 3.2.2. We then describe the process of training NMT model in Section 3.2.3. Finally, we show how *AccuSpell* corrects a given sentence at run-time by applying NMT model in Section 3.3.

3.2 Learning to Correct Misspelled Sentence

We attempt to train a neural machine translation (NMT) model using right-and-wrong sentence pairs from edit logs and artificial data, which to translate a misspelled sentence into a correct one. In this training process, we first extract the sentences with spelling errors from edit logs (Section 3.2.1) and generate artificial misspelled sentences from a set of error-free sentences (Section 3.2.2). We then use these data to train the NMT model (Section 3.2.3).

3.2.1 Extracting Misspelled Sentences from Edit Logs

In the first stage of training process, we extract a set of sentences with spelling errors annotated by simple edit tags (i.e., <[-, -]> for deletion and <{+, +} > for insertion). For example, the sentence “希望未來主要島嶼都有完善的[-馬-]{+碼+}頭，” (Hope that the main islands will have perfect docks in the future.) contains the edit tags “[-馬-]{+碼+}” that means the original character “馬” (pronounced ’ma’) was replaced with “碼”

(pronounced 'ma').

【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

1. 食藥署**[-今-]{+昨+}**宣布將開放食鹽添加微量氟化物，
2. 現在「**[-b-]{+B+}**lue Monday變**[-g-]{+G+}**reen Monday了，
3. 記得拍照帶走美景就好**{+。+}[-；-]**如果時間許可，
4. **[-她-]{+他+}**昨日接受專訪時說明，
5. 參加世界盃拔河賽獲4金2銀的大笨牛夢想拔河隊教練陳**[-鵬-]{+建+}**文，
6. 最後再撒上適量起**{+司+}[-士-]**絲，
7. 這項計畫將持續募款到今年**[-聖-]{+耶+}**誕節，
8. 使得泰山今年**[-上-]{+下+}**半年獲利成長樂觀。
9. 饗蔬職人**[-除了提供全素（非奶蛋素）的新鮮食材外，-]**還用心烹調3種湯頭、7種醬料，
10. 價值上百萬的好禮**[-通通-]{+統統+}**帶回家。
11. 希望未來主要島嶼都有完善的**[-馬-]{+碼+}**頭，

Figure 3. Examples of different edit types in edit logs

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，” (”Foreign investment is not in a hurry to layout next year,”).

After that, we attempt to extract the sentences that contain at least one typo. As shown in Figure 3, the edit logs could contain 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 errors. 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. Furthermore, we use a set of rules to filter out some kinds of edits such as time-related and digital-related. Figure 3 shows some edited sentences, the fifth, sixth, seventh, eighth and eleventh sentences are regarded as sentences with spelling errors according these simple rules. The output of this stage is a set of sentences with spelling errors annotated using simple edit tags, as shown in Figure 4.

- 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 4. Example outputs for the step of extracting misspelled sentences

Although this approach for extracting the edited sentences involving spelling correction can obtain quite a few results, there is still a room for improvement. For example, the edited sentence “價值上百萬的好禮[-通通-]{+統統+}帶回家。” (’Bring millions of good gifts home’) contains a consecutive two-character edit pair “[-通通-]{+統統+}” (both pronounced ’tong tong’), which is also spelling error correction. However, it is not extracted because we only consider consecutive single-character edit pairs. In some cases, an edited sentence might be wrongly regarded as misspelled sentence. For example, the sentence “這

項計畫將持續募款到今年[-聖-]{+耶+}誕節，” (’This project will continue to raise funds until this Christmas,’) contains an edit pair “[-聖-]{+耶+}” about style modification. Consider the context of the edited character, the word “聖誕節” (pronounced ’sheng dan jie’, it means the birthday of the holy child Jesus) and “耶誕節” (pronounced ’ye dan jie’, it means the birthday of Jesus) are both correct, and they almost mean the same thing. For such case, using word segmentation and meaning similarity measure of two words may be helpful.

3.2.2 Generating Artificially Misspelled Sentences

In the second stage of training process, we create a set of artificial misspelled sentences for expanding our training data. These generated data are expected to make the Chinese spelling checker more effective.

```
procedure GenerateErrorSentence_Map(CorrectSentences)

    for each Sentence in CorrectSentences
       for each Word_i in Sentence
(1)       WrongWords = getConfusionSet(Word_i)
          for each WrongWord_j in WrongWords
(2a)         WrongSentence = Sentence
(2b)         replace WrongSentence_i with WrongWord_j
(3a)         WordPair = Word_i + "|||" + WrongWord_j
(3b)         SentencePair = Sentence + "|||" + WrongSentence
(4)          output <WordPair, SentencePair>


procedure GenerateErrorSentence_Reduce(WordPairs, SentencePairs)

(1) N = n
    for each WordPairs, SentencePairs
(2)    shuffle SentencePairs
(3)    output top N SentencePairs
```

Figure 5. Generating artificial misspelled sentence

Table 1. Examples of confusion set

Correct Word	Wrong Words
部署(’arrange’, pronounced ’bu shu’)	布署, 部處, 佈署, 步署
賠罪(’apologize’, pronounced ’pei zui’)	培罪, 陪罪

The input to this stage is a set of presumably error-free sentences from published texts with word segmentation done using a word segmentation tool provided by the CKIP Project (Ma & 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" ('also apologized to the patient for ten minutes') to generate three misspelled sentences, as shown in Table 2. Figure 5 shows the procedure for generating artificial misspelled sentences using the MapReduce framework to speed up the process.

Table 2. Artificial misspelled sentences for '也跟患者賠罪了十分鐘'

Artificial Misspelled Sentence	Replaced Word	Wrong Word
也跟患者培罪了十分鐘	賠罪	培罪
也跟患者陪罪了十分鐘	賠罪	陪罪
也跟患者賠罪了十分鍾	分鐘	分鍾

- **Map procedure**: In Step (1), for each word in the given (presumably) error-free sentence with length not longer than 20 words, we obtain the corresponding confused words. For example, the confusion set of word "賠罪" contains two confused wrong words: "培罪" and "陪罪". The original word is then replaced with its corresponding confused words in Steps (2a) and (2b). To work with *MapReduce* framework, we then format the output data to key-value pair in Step (3a) and (3b). In order to group the generated misspelled sentences according to replacement (e.g., "賠罪" is replaced with "培罪"), we use a right-and-wrong word pair (e.g., "賠罪|||培罪") to be the key, and a right-and-wrong sentence pair (e.g., "也跟患者賠罪了十分鐘|||也跟患者培罪了十分鐘") to be the value. Finally, the key-value pair is outputted in Step (4).
- **Reduce procedure**: In this procedure, the inputs are the key-value pairs outputted by Mapper. For each word pair, there might be too many sentence pairs. Thus, in Step (1), we set a threshold N to limit the number of sentences generated. In order to randomly sample a set of sentences, we make these sentence pairs redistributed by shuffling in Step (2), and output the first N of sentence pairs in Step (3).

The output of this stage is a set of right-and-wrong sentence pairs, as shown in Table 3.

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.

Table 3. Example outputs for the step of generating artificial misspelled sentences

Right Sentence	**Wrong Sentence**
可見酒精會讓白老鼠上癮，	可見酒精會讓白老鼠上蔭，
導致水圳混濁不堪，	導致水圳混濁不勘，
媒體何嘗沒有一點責任？	媒體何賞沒有一點責任？
地處偏僻且巷弄狹窄，	地處編僻且巷弄狹窄，
希望他的覺醒為時不晚。	希望他的覺省為時不晚。

3.2.3 Neural Machine Translation Model

In the third and final stage of training process, 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 $X = [x_1,x_2,...,x_I]$ and the decoder generates translated target sentence $Y = [y_1,y_2,...,y_J]$. 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 corrected. To train the NMT model, we use a set of right-and-wrong sentence pairs from edit logs (Section 3.2.1) and artificially-generated data (Section 3.2.2) 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 $\mathbf{e} = [e_1,e_2,...,e_I]$, 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 $\{y_1, y_2, \ldots, y_{j-1}\}$:

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

The conditional probability is modeled as:

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

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

We use an 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:

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

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, Cho, and Bengio (2014). We use the global attention approach (Luong, Pham & Manning, 2015) 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, Kim, Deng, Senellart, & Rush, 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.

3.3 Run-time Error Correction

Once the NMT model is automatically trained for correcting spelling errors, we apply the model at run time. *AccuSpell* then corrects a given potentially misspelled sentence with the character-based NMT model using the procedure in Figure 6.

```
procedure CorrectSpellingError(Sentence)

(1)  sourceSentence = tokenize(Sentence)
(2)  targetSentence = NMTModel(sourceSentence)

(3a) copy sourceSentence to Result
     for each sourceChar_i in sourceSentence:
        if sourceChar_i not equals to targetChar_i
(3b)       replace Result_i with "[-sourceChar_i-]{+targetChar_i+}"

(4)  return Result
```

Figure 6. Correcting spelling errors in a sentence

With a character-based NMT model, the input sentence is expected to follow the format that tokens are space-separated. Thus, in Step (1), the characters in the given sentence are separated with space. For example, “今晚月色很美，我想小灼一杯。” is transformed into “今晚月色很美，我想小灼一杯。”. In Step (2), the source sentence is fed to our NMT model. During processing, the encoder first transforms the source sentence into a sequence of vectors. The decoder then computes the probabilities of predicted target sentences given the vectors of source sentence. Finally, a beam search is used to find a target sentence that approximately maximizes the conditional probability. Table 4 shows the top three target sentences predicted by our NMT model for the source sentence “今晚月色很美，我想小灼一杯。”, and the highest-score one “今晚月色很美，我想小酌一杯。” is returned as the correction.

Table 4. Top three target sentences of the source sentence "今晚月色很美，我想小灼一杯。" predicted by NMT model

Target Sentence	Predicted Score	Rank
今晚月色很美，我想小酌一杯。	-0.0047	1
今晚月色也美，我想小酌一杯。	-6.93	2
今晚月色很美，我想小灼一耶。	-7.36	3

To give useful and clear feedback, we convert the correction result into a informative expression instead present users with the output of NMT model directly. Therefore, in Steps (3a) and (3b), we compare the source sentence with the target sentence to find out the differences between them, and use simple edit tags to mark these differences. Finally in Step (4), the converted result (e.g., "今晚月色很美，我想小[-灼-]{+酌+}一杯。") is returned by *AccuSpell*. As shown in Figure 1, the characters to be deleted (e.g., "[-灼-]") are colored in red, while the inserted characters (e.g., "{+酌+}") are colored in green.

4. Experimental Setting

AccuSpell was designed to correct spelling errors in Chinese texts written by native speakers. As such, *AccuSpell* will be trained and evaluated using mainly real edit logs and a newspaper corpus. 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. However, lack of edit type annotation makes it difficult to directly identify spelling errors. Thus, we extracted a set of annotated sentences involving spelling error correction from this edit logs using the approach described in Section 3.2.1. To train on NMT model, we transformed every annotated sentence into a source-and-target parallel sentence. For example, "外資也不急著[-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. The 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 3.2.2.

Table 5. Examples of 聯合報統一用字(Uniform Words List of UDN)

	Recommended word	**Unrecommended word**
巴吧 (pronounced 'ba')	啞巴('dumb')	啞吧
背揹 (pronounced 'bei')	背著('carrying') 背黑鍋('take the blame')	揹著 揹黑鍋
刨鉋 (pronounced 'bao')	刨冰('shaved ice')	鉋冰
杯盃 (pronounced 'bei')	市長杯('mayor cup')	市長盃
澹淡 (pronounced 'dan')	慘澹('miserable') 淡泊 ('indifferent')	慘淡 澹泊
闆板 (pronounced 'ban')	老闆('boss')	老板

Confusion Set: We used five distinct confusion sets collected from different sources:

- **聯合報統一用字(Uniform Words List of UDN):** The dataset of 聯合報統一用字 provided by UDN Digital contains 1,056 easily confused word pairs. As shown in Table 5, the confused word pairs indicate that which words are recommended and which ones should not be used for UDN news articles. However, not all the unrecommended words are wrong because the suggestions are just preference rules for writing news articles for the UDN journalists. For example, a confused word pair [“市長杯”, “市長盃”]('Mayor CUP') in Table 5, the former is recommended and the latter is not recommended, but they are both correct and in common use. In our work, we collect all the word pairs, and consider them as right-and-wrong word pairs
- **東東錯別字(Kwuntung Typos Dictionary):** This dataset was collected from the Web (www.kwuntung.net/check/), which contains a set of commonly confused right-and-wrong word pairs. For each word pair, there is one distinct character with similar pronunciation or shape between right and wrong word. We obtain 38,125 different right-and-wrong word pairs in total, which constitutes the main part of our confusion set.
- **新編常用錯別字門診(New Common Typos Diagnosis):** This dataset comes from the print publication: 新編錯別字門診 (蔡有秩, 2003) and contains 492 right-and-wrong

word pairs.

- **常見錯別字辨正辭典(Dictionary of Common Typos):** This dataset is from a print publication: 常見錯別字辨正辭典 (蔡榮圳, 2012). There are 601 right-andwrong word pairs in total.
- **國中錯字表(The Typos List for Middle School):** This dataset contains a set of commonly misused right-and-wrong word pairs for middle school students. There are 1,720 word pairs in original. However, some pairs are composed of phrases (e.g., “觀念不佳” and “為自己的未來鋪路”) instead of words. To ensure that all pairs are at word level, we used some rules to transform the phrase pairs into word pairs. For example, the right-and-wrong phrase pair [“為自己的未來鋪路”, “為自己的未來捕路”] (’Pave the way for your own future’) is transformed to the word pair [“鋪路”, “捕路”] (pronounced ’pu lu’ and ’bu lu’). Moreover, we discarded the pairs cannot be transformed such as [“十來枝的掃具”, “十來隻的掃具”] (’A dozen brooms.’). After that, 1,551 word pairs remained.

The confused word pairs of five confusion sets 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”](’Clock’, pronounced ’zhong biao’). “鐘錶” is right and “鐘表” is wrong in Kwuntung Typos Dictionary, while “鐘表” is adopted and “鐘錶” is not recommended in Uniform Words List of UDN. 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, the new confusion set **CFset** with 33,551 distinct commonly confused word pairs were obtained. Table 6 shows the number of word pairs of all confusion sets.

Table 6. Number of word pairs of five confusion sets

Confusion Set	Number of confused word pairs
Uniform Words List of UDN	1,056
Kwuntung Typos Dictionary	38,125
New Common Typos Diagnosis	492
Dictionary of Common Typos	601
The Typos List for Middle School	1,460
CFset	33,551

Table 7. The statistics of test sets

	UDN Edit Logs	SIGHAN-7
# of sentences	1,175	6,101
# of sentences with errors	919	1,222
# of sentences without errors	256	4,879
# of error characters	919	1,266
Average # of errors in sentences with errors	1	1.04
Average length of sentences	17.47	12.16

Test Data: We used two test sets for evaluation, and Table 7 shows the statistical analysis of them in detail:

- **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, Liu & Lee, 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 be consistent with UDN Edit Logs, we segmented these sentences into 6,101 clauses, and 1,222 of which contain at least one error.

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 2layer LSTM. Both the encoder and the decoder have 500 hidden units. We use the Adam Algorithm (Kingma & 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.

4.3 Models Compared

Our experimental evaluation focuses on writing of native speakers. Therefore, we used UDN Edit Logs and the artificially generated misspelled sentences as the training data. To investigate whether adding artificially generated data improves the performance of our Chinese spelling check system, we compared the results produced by several models trained on different combination of datasets.

In addition, we use some additional features on source and target words in the form of discrete labels to train the NMT model[1]. As Liu *et al*. (2011) stated, around 75% of typos were related to the phonological similarity between the correct and the incorrect characters, and about 45% were due to visual similarity. Thus, we use the pronunciation and shape of a character from the Unihan Database[2] as the additional feature of the source and target characters. As an example, for the character “詣”, the pronunciation feature is “ㄧ” (without considering the tone) and the shape features are “言” and “旨”. On the other hand, a spelling error might involve not only the character itself but also the context, so we use the context (with window size 1) of a character as additional features to train another model.

Table 8. Features for the sentence “我想小酌一杯。”

Feature	我	想	小	酌	一	杯	。
Sound	ㄨㄛ (wo)	ㄒㄧㄤ (xiang)	ㄒㄧㄠ (xiao)	ㄓㄨㄛ (zhuo)	ㄧ (yi)	ㄅㄟ (bei)	N
Shape	(戈,我)	(心,相)	(小,小)	(酉,勺)	(一,一)	(木,不)	(N,N)
Context	(BEG,想)	(我,小)	(想,酌)	(小,一)	(酌,杯)	(一,。)	(杯,END)

Table 8 gives an example to illustrate the pronunciation, shape, and context features.

There are totally eight models trained for comparing, and only last two were trained with features. The eight models evaluated and compared are as follows:

- **UDN-only:** The model was trained on 226,913 sentence pairs from the training part of UDN Edit Logs.
- **UDN + Artificial (1:1):** The model was trained on 226,913 sentence pairs from the training part of UDN Edit Logs plus 225,985 artificially generated sentence pairs (452,871 in total).
- **UDN + Artificial (1:2):** The model was trained on 226,913 sentence pairs from the training part of UDN Edit Logs plus 440,143 artificially generated sentence pairs (667,056

[1] https://opennmt.net/OpenNMT/data/word_features/
[2] http://www.unicode.org/charts/unihan.html

in total).

- **UDN + Artificial (1:3):** The model was trained on 226,913 sentence pairs from the training part of UDN Edit Logs plus 673,006 artificially generated sentence pairs (899,919 in total).
- **UDN + Artificial (1:4):** The model was trained on 226,913 sentence pairs from the training part of UDN Edit Logs plus 899,385 artificially generated sentence pairs (1,126,298 in total).
- **Artificial-only:** The model was trained on 899,385 artificially generated sentence pairs.
- **FEAT-Sound & Shape:** The model was trained on the same data in *UDN +Artificial (1:3)* model with pronunciation and shape of character features.
- **FEAT-Context:** The model was trained on the same data in *UDN + Artificial (1:3)* model with context features.

4.4 Evaluation Metrics

Chinese spelling check systems are usually compared based on two main metrics, precision and recall. We use the metrics provided by SIGHAN-8 Bake-off 2015 for Chinese spelling check shared task (Tseng, Lee, Chang, & Chen, 2015), which include False Positive Rate, Accuracy, Precision, Recall, and F1, to evaluate our systems.

The confusion matrix is used for calculating these evaluation metrics. In the matrix, TP (True Positive) is the number of sentences with spelling errors that are correctly identified by the developed system; FP (False Positive) is the number of sentences in which non-existent errors are identified; TN (True Negative) is the number of sentences without spelling errors which are correctly identified as such; FN (False Negative) is the number of sentences with spelling errors that are not correctly identified. The following metrics are calculated using TP, FP, TN and FN:

$$False\ Positive\ Rate\ (FPR) = \frac{FP}{FP + TN} \tag{12}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{16}$$

Table 9. The given test sentences with gold standards

Sentence ID	Sentence	Gold Standard
S1	希望藉此鼓勵自己和他人要積極樂觀實現夢想。	0
S2	PM2.5 對人體健康為害大，	11, 危
S3	因為難以達到連數門檻，	8, 署
S4	他仍記得自已當年還是學校棒球隊員，	6, 己
S5	剛推動的社會住密也要設一定比例的大陽光電。	8, 宅, 17, 太
S6	美麗的勇士山頭將被掏空了嗎？	10, 淘
S7	未來發展需要新的能力、新的動能，	0
S8	學生因宗教、重族、國籍而遭羞辱者大幅增加。	7, 種

Table 10. The results outputted by the system

Sentence ID	Output Sentence	Correction
S1	希望藉此鼓勵自己和他人要積極樂觀實現夢想。	0
S2	PM2.5 對人體健康危害大，	11, 危
S3	因為難以達到連數門檻，	8, 署
S4	他還記得自己當年還是學校棒球隊員，	2, 還, 6, 己
S5	剛推動的社會住宅也要設一定比例的大陽光電。	8, 宅
S6	美麗的勇士山頭將被掏空了嗎？	0
S7	未來發展需要新的能力、新的動能，	15, 力
S8	學生因宗教、種族、國籍而遭羞辱者大幅增加。	7, 種

For example, given 8 test sentences with gold standards shown in Table 9. Assume that our system outputs the results as shown in Table 10, the evaluation metrics will be measured as follows:

- FPR = 0.5 (= 1/2)

  Notes: {S7}/{S1, S7}

- Accuracy = 0.5 (= 4/8)

  Notes: {S1, S2, S3, S8}/{S1, S2, S3, S4, S5, S6, S7, S8}
- Precision = 0.5 (= 3/6)

  Notes: {S2, S3, S8}/{S2, S3, S4, S5, S7, S8}
- Recall = 0.75 (= 3/4)

  Notes: {S2, S3, S8}/{S2, S3, S6, S8}
- F1 = 0.6 (= 2*0.5*0.75/(0.5+0.75))

5. Results and Discussion

In this section, we report the results of experimental evaluation using the resources and metrics described in previous chapter. Specifically, we report the results of our evaluation, which contains two test sets evaluated by false positive rate (FPR), accuracy, precision, recall, and F1 score. First, we present the results of several models evaluated on two test sets in Section 5.1. We then give some analysis and discussion of the errors in the two test sets in Section 5.2.

5.1 Evaluation Results

Table 11 shows the evaluation results of UDN Edit Logs. As we can see, all models trained on edit logs and artificially generated data perform better than the one trained on only edit logs. Moreover, the model trained on only edit logs performs slightly worse, while the model trained on only artificially generated data performs the very worst on all metrics. Even though the model trained with sound and shape features performs relatively poorly on FPR, it has the best performance on accuracy, precision, recall, and F1 score.

Table 11. Evaluation results of UDN Edit Logs

Model	FPR	Accuracy	Precision	Recall	F1
UDN-only	.066	.64	.80	.64	.71
UDN + Artificial (1:1)	.090	.69	.84	.69	.76
UDN + Artificial (1:2)	.063	.71	.86	**.72**	.78
UDN + Artificial (1:3)	.066	.70	.86	.69	.76
UDN + Artificial (1:4)	**.059**	.71	.87	.71	.78
Artificial-only	.137	.35	.43	.26	.33
FEAT-Sound & Shape	.098	**.72**	**.88**	**.72**	**.79**
FEAT-Context	.059	.71	.87	.70	.78

Table 12. Evaluation results of SIGHAN-7

Model	FPR	Accuracy	Precision	Recall	F1
UDN-only	.109	.74	.19	.17	.18
UDN + Artificial (1:1)	.089	.83	.50	.59	.54
UDN + Artificial (1:2)	.081	.84	.54	.61	.57
UDN + Artificial (1:3)	.078	**.85**	.56	.62	.58
UDN + Artificial (1:4)	**.073**	**.85**	**.58**	.63	**.61**
Artificial-only	.079	.84	.53	.58	.56
FEAT-Sound & Shape	.097	.83	.51	**.64**	.57
FEAT-Context	.080	.84	.56	.61	.58

For the other test set, SIGHAN-7, the evaluation results are shown in Table 12. UDN + Artificial (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 significantly worse performance than others, 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 additional features of either sound and shape or context.

In general, we obtain extremely low average FPR evaluated on the two test sets. There are three obvious differences between the results of two test sets. First, the model trained on only edit logs (**UDN-only**) and the model trained on only artificially generated data (**Artificial-only**) have the opposite results on UDN Edit Logs and SIGHAN-7. As we can see, **UDN-only** performs well on UDN Edit Logs but very poorly on SIGHAN-7. In contrast, **Artificial-only** has worst performance on UDN Edit Logs but acceptable performance on SIGHAN-7. Second, we obtain relatively high precision compared with recall on UDN Edit Logs, while higher recall than precision on SIGHAN-7. Third, in Table 13, 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.

Table 13. Evaluation results related to the models trained with features

Test Set	Model	FPR	Accuracy	Precision	Recall	F1
UDN Edit Logs	UDN + Artificial (1:3)	.066	.70	.86	.69	.76
	FEAT-Sound & Shape	.098	**.72**	**.88**	**.72**	**.79**
	FEAT-Context	**.059**	.71	.87	.70	.78
SIGHAN-7	UDN + Artificial (1:3)	**.078**	**.85**	**.56**	.62	**.58**
	FEAT-Sound & Shape	.097	.83	.51	**.64**	.57
	FEAT-Context	.080	.84	**.56**	.61	**.58**

Table 14. Distribution of the relations between typos and corrections in test sets

	UDN Edit Logs	SIGHAN-7
# of error characters	919	1,266
Similar Sound	70%	84%
Similar Shape	36%	40%
Similar Sound and Shape	30%	30%

5.2 Error Analysis

The nature of our two test sets are different, UDN Edit Logs are produced by newspaper editors, while SIGHAN-7 are collected from essays written by junior high students. Therefore, we analyze and discuss the details of the two test sets in this section.

We use the confusion sets provided by SIGHAN 7 Bake-off 2013 (Wu *et al.*, 2013), which contains a set of characters with similar pronunciation and shape, to analyze the relations between typos and the corresponding corrections in our test data. There are 919 typos in UDN Edit Logs and 1,266 typos in SIGHAN-7. As shown in Table 14, the analysis results of UDN Edit Logs and SIGHAN-7 are similar. Most of typos are related to similar pronunciation, and over 35% of typos are due to similar shape. Moreover, around 30% of typos are associated with similar pronunciation as well as shape.

Table 15 and 16 show some analysis of evaluation results of UDN Edit Logs and SIGHAN-7 respectively. As we can see, according to the analysis of the errors which were not corrected by models, there is no significant difference among these different models. In both UDN Edit Logs and SIGHAN-7, around half of the spelling errors not corrected are related to similar pronunciation no matter which model we used.

Table 15. Distribution of the relations between not corrected typos and corrections of the evaluation results using UDN Edit Logs

Model	# of errors not corrected	Similar Sound	Similar Shape	Similar Sound and Shape
UDN-only	404	52%	7%	27%
UDN+Artificial (1:3)	340	54%	8%	26%
Artificial-only	733	43%	6%	26%
FEAT-Sound&Shape	299	57%	8%	25%

Table 16. Distribution of the relations between not corrected typos and corrections of the evaluation results using SIGHAN-7

Model	# of errors not corrected	Similar Sound	Similar Shape	Similar Sound and Shape
UDN-only	1,092	57%	9%	27%
UDN+Artificial (1:3)	596	60%	8%	22%
Artificial-only	641	58%	8%	24%
FEAT-Sound&Shape	597	58%	8%	24%

It is worth discussing that there are some special cases in the test sets. For example, an error character “怖” (pronounced ’bu’) occurring in some words such as “怖告欄” (pronounced ’bu gao lan’) and “怖置” (pronounced ’bu zhi’) should be corrected to “佈” (pronounced ’bu’) in SIGHAN-7. However, the correction predicted by our models is “布” since we used the Chinese dictionary published by Ministry of Education of Taiwan as the gold standards of our training data. According to the dictionary, “佈置” and “佈告欄” are invalid, while “布置” (’decorate’) and “布告欄” (’bulletin board’) are legal. Another case is related to grammatical errors. Our models aim to correct spelling errors, but there are some sentences with grammatical errors in SIGHAN-7 such as “要如何在站起來呢？” (’How to stand up again?’) and “哪激的起美麗的浪花？” (How can it stir up the beautiful spray?), where “在” (pronounced ’zai’) and “的” (pronounced ’de’) should be “再” (pronounced ’zai’) and “得” (pronounced ’de’) respectively. These kinds of errors are involved the dependency structure of sentences. In the predicted results of our models, we found that the model trained on only artificially generated data cannot correct such errors. Other models using edit logs have slightly better performance on correcting these kinds of errors, but there isn’t too much of a difference.

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” (pronounced ’zhi’), which is not corrected by a model trained without features because our training data does not cover this typo. However, the sentence is correctly translated into “他在文學方面有很高的造詣。” by the model trained with sound and shape features.

6. Conclusion and Future Work

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 and methods of augmenting data for CSC. We collect a series of confusion set from the Web, including 東東錯別字 (Kwuntung Typos Dictionary), 新編常用錯別字門診(New Common Typos Diagnosis), 常用錯別字(Dictionary of Common Typos), 國中錯字表(The Typos List for Middle School). To augment more data for training an NMT model, we develop a way of injecting artificial errors into error-free sentences with the confusion sets. In addition, we compare the different ratio of mixture of real and artificial data and more artificial data improves the performance. Finally, we conduct experiments on models with additional features (e.g., pronunciation, shape components, and context words) to show that phonological, visual, and context information can improve the recall and reveal the ability to generalize common typos.

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 neural machine translation 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.

References

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. In arXiv preprint arXiv:1409.0473.

Chang, C.-H. (1995). A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, *95*, 278-283.

Chiu, H.-w., Wu, J.-c., & Chang, J. S. (2013). Chinese spelling checker based on statistical machine translation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 49-53.

Chollampatt, S. & Ng, H. T. (2018). A multilayer convolutional encoder-decoder neural network for grammatical error correction. In arXiv preprint arXiv:1801.08831.

Felice, M. & Yuan, Z. (2014). Generating artificial errors for grammatical error correction. In Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, 116-126. doi: 10.3115/v1/E14-3013

Gu, S. & Lang, F. (2017). A chinese text corrector based on seq2seq model. In *Proceedings of 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 322-325. doi: 10.1109/CyberC.2017.82

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. In arXiv preprint arXiv:1412.6980.

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). Opennmt: Opensource toolkit for neural machine translation. In arXiv preprint arXiv:1701.02810.

Liu, C.-L., Lai, M.-H., Tien, K.-W., Chuang, Y.-H., Wu, S.-H., & Lee, C.-Y. (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. doi: 10.1145/1967293.1967297

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attentionbased neural machine translation. In arXiv preprint arXiv:1508.04025.

Ma, W.-Y. & Chen, K.-J. (2003). Introduction to ckip chinese word segmentation system for the first international chinese word segmentation bakeoff. In *Proceedings of the 2nd SIGHAN on CLP*, 168-171. doi: 10.3115/1119250.1119276

Rei, M., Felice, M., Yuan, Z., and Briscoe, T. (2017). Artificial error generation with machine translation and syntactic patterns. In arXiv preprint arXiv:1707.05236.

Tseng, Y.-H., Lee, L.-H., Chang, L.-P., & Chen, H.-H. (2015). Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, 32-37. doi: 10.18653/v1/W15-3106

Wu, S.-H., Chen, Y.-Z., Yang, P.-C., Ku, T., & Liu, C.-L. (2010). Reducing the false alarm rate of chinese character error detection and correction. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*.

Wu, S.-H., Liu, C.-L., & Lee, L.-H. (2013). Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 35-42.

Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., & Ng, A. Y. (2016). Neural language correction with character-based attention. In arXiv preprint arXiv:1603.09727.

Yuan, Z. & Briscoe, T. (2016). 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, 380-386. doi: 10.18653/v1/N16-1042

Zhang, L., Huang, C., Zhou, M., & Pan, H. (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*, 248-254. doi: 10.3115/1075218.1075250

蔡有秩 (2003)。新編錯別字門診。 語文訓練叢書，螢火蟲。[Tsai, Y.-J. (2003). *New Common Typos Diagnosis*, Fireflybooks.]

蔡榮圳 (2012)。常見錯別字辨正辭典。中文可以更好，商周出版。 [Tsai, R.-J. (2012). *Dictionary of Common Typos*, Business Weekly.]