applied sciences

MDPI

Article

Think Twice: A Post-Processing Approach for the Chinese Spelling Error Correction

Wei Gou and Zheng Chen *

School of Information and Software Engineering, University of Electronic Science and Technology, Chengdu 610051, China; 201852090720@std.uestc.edu.cn

* Correspondence: zchen@uestc.edu.cn; Tel.: +86-1860-281-5226

Abstract: Chinese Spelling Error Correction is a hot subject in the field of natural language processing. Researchers have already produced many great solutions, from the initial rule-based solution to the current deep learning method. At present, SpellGCN, proposed by Alibaba's team, achieves the best results of which character level precision over SIGHAN2013 is 98.4%. However, when we apply this algorithm to practical error correction tasks, it produces many false error correction results. We believe that this is because the corpus used for model training contains significantly more errors than the text used for model correcting. In response to this problem, we propose performing a post-processing operation on the error correction tasks. We employ the initial model's output as a candidate character, obtain various features of the character itself and its context, and then use a classification model to filter the initial model's false error correction results. The post-processing idea introduced in this paper can apply to most Chinese Spelling Error Correction models to improve their performance over practical error correction tasks.

Keywords: natural language processing; Chinese Spelling Error Correction; post-processing; practical application

Citation: Gou, W.; Chen, Z. Think Twice: A Post-Processing Approach for the Chinese Spelling Error Correction. *Appl. Sci.* **2021**, *11*, 5832. https://doi.org/10.3390/app11135832

Academic Editor: Rafael Valencia-García

Received: 8 April 2021
Accepted: 15 June 2021
Published: 23 June 2021

1. Introduction

Errors will inevitably occur in the text entered by people. Therefore, researchers have proposed a large number of error correction algorithms to minimize errors in text. As an ancient writing system, the Chinese have tens of thousands of different characters, most of which are graphic variants. However, the Chinese input methods in the modern age are mostly based on Pinyin. As a result, Chinese spelling errors include two different error patterns: phonological errors and visual errors. Compared to English, Chinese text has no character boundaries, and the diversity of errors bring significant challenges to Chinese Spelling Error Correction (CSC).

The language model plays an essential role in Chinese text error correction algorithms. In the early years, CSC mostly utilized n-gram language models or character vector models, until the introduction of Bidirectional Encoder Representations from Transformers (BERT [1]) in 2018, which provided a strong base-line for the CSC. Since then, various BERT-based algorithms, such as FASPell [2], Soft-Mask [3], and SpellGCN [4], have kept pushing the CSC's state of the art to a new high record. According to its paper, SpellGCN achieves 98.4 precision, 88.4 recall, and 93.1 F1 score over the SIGHAN 2013 dataset [5]. Encouraged by this result, we decided to apply SpellGCN to one of our online text editing systems. However, its actual performance was not as good as expected. The model replaces a massive number of correct characters with characters that it believes to be more reasonable, which is not expected in the error correction system.

CSC models are trained on a specific CSC corpus, which contains more errors than our daily texts. Therefore, in the error correction stage, the model may have a false estimate of the text's error rate (the percentage of error sentences in the text), resulting in a greater likelihood of changing each input character. On the other hand, the model is a simple

reflection model that chooses the character with the highest probability in the last layer as output, without any further consideration. However, there can be many suitable characters in every slot in a Chinese sentence. A correct output may not be a necessary correction. How to make the model avoid these fault correction in the practical application becomes very important.

Inspired by post-editing in machine translation, we propose a post-processing operation for the Chinese Spelling Error Correction task. We employ the state-of-the-art CSC model SpellGCN as the first step, use its output as candidate characters, and then use the second model to decide whether to adopt the candidate error correction. The post-processing model is a simple classification model, which uses various features of the candidate character and original character as input, such as topP, topK, pro, rank, etc. We also consider its context features, such as uncertainty and similarity. By employing the post-processing operation, the CSC system can significantly improve the error correction precision, but at the cost of slightly reducing the recall. The overall F1 score can be increased by 10% or more. Researchers can apply our approach to various models that rely on probability for output selection to improve its performance over practical error correction tasks.

2. Related Work

In the early days, researchers adopted the idea of combining rules and language models to solve error correction in Chinese texts. They relied on confusion sets to modify common errors in sentences. Each element in the confusion set is a dictionary. The key is an error-prone Chinese character or word, and the corresponding values contain a series of characters or phrases similar to the key in sound or form. In error processing, the sentence is segmented into many characters and words. Researchers chose one of the values to replace the error if the confusion set included the element corresponding to these characters or words. After that, they used the language model to score the replaced sentences and then selected the sentence with the highest score as the error correction result [6,7]. However, typos affected the segmentation results in the sentence and reduced the effect of error correction. Chiu and Wu proposed the noisy channel model [8] to address this limitation. Zhong and Wang utilized this phenomenon that a misspelled word is often split into two or more characters, and used the Single Source Shortest Path (SSSP) algorithm to correct Chinese spelling errors [9]. In Yang and Zhao's work, they used Minimized-Path Segmentation [10] to solve this problem.

With the development of deep learning research and the continuous improvement of computing power, researchers began to use neural networks to solve Chinese error correction problems. Qiu and Qu used two stages to solve the CSC problem in their research. First, they used the confusion set and language model to determine the suspected error position in the sentence and then used the Sequence-to-Sequence [11] model to correct the mistakes [12]. At the same time, Duan and Wang combined the Long Short Term Mermory network (LSTM [13]) network with the Conditional Random Fields(CRF [14]) to accomplish Chinese text error detection through sequence labeling [15]. After that, they incorporated the Sequence-to-Sequence model and attention to realize the error correction of Chinese text based on Bi-LSTM [16] model [17]. Lu Xie and Li introduced the pronunciation and structure of words as model inputs and then combined Bi-LSTM and CRF to find errors in sentences. After that, they used the Masked Language model to correct suspected errors [18]. On the basis of the original LSTM model, Wang and Duan added the pronunciation and confusion set information to the model by introducing the fusion cell form to help realize text error correction [19]. Wang and Liu rethought the attention mechanism [20], combined the confusion set and regard the Chinese error correction task as a prediction problem. They introduced more global information as a new attention mechanism distribution and used LSTM to predict the target character [21].

More recently, pretrained language models have played a significant role in achieving impressive gains in a wide varity of Natural language processing tasks. These models

are trained on a large amount of unsupervised corpora to learn sufficient corresponding semantic knowledge. Fine-tuning these models on the professional corpus can obtain good performance while alleviating the problem of insufficient labeled corpora. Bidirectional Encoder Representations from Transformers (BERT) [1], proposed by the Google team, achieved leading results in various natural language processing tasks once it came out. Therefore, researchers began to apply BERT and its variant models to Chinese Spelling Error Correction tasks. The ByteDance team eliminated the soft-mask mechanism. They used Bi-GRU to softly cover the error and corrected whole sentences by the Masked Language model [3]. The Alibaba team introduced the Chinese character information's glyph and phonetic information into the BERT model's uppermost layer and expanded the training data. This algorithm obtained new, state-of-the-art results for error correction tasks [4].

Considering Chinese characters' particularity, more and more researchers have begun to pay attention to Chinese characters' phonetic and morphological information. They added this information to the CSC algorithm in different ways. Han and Wang et al. still treated the Chinese error detection task as a sequence tagging task based on the Bi-LSTM [16]. They considered the pronunciation and glyph structure of Chinese characters as the model's input, which significantly improved the model's error detection ability [22]. Wang and Zhong et al. used the word-embedding method as the judgment reliance of the model prediction [23]. After the original model's outputted results, the iQiyi's team filtered those results by artificially determining the filtering curve [21]. This method was similar to our idea, but the filtering curve needed to be artificially selected, so the workload was massive [2]. Nguyen and Ngo et al. proposed the adaptable filter, which uses hierarchical embeddings [24] to filter the suggested amendments given by the Masked Language model.

3. Motivation

In the research field of Chinese Spelling Error Correction, we usually test the algorithm's capabilities on some public datasets, which contain many errors (such as SIGHAN [5,25,26]). So far, the Alibaba team proposed SpellGCN, which obtained the SOTA F1 score on SIGHAN13-15 (the character level correction F1 scores are respectively 93.1%, 85.6%, and 89.4%). Unexpectedly, when we applied this model to correct erroneous texts in reality, it produced many unnecessary corrections. After careful analysis of the wrong samples, we believe that the difference of text error rate (the percentage of error sentences in the text) caused this result. In machine learning theory, the Independent and Identically Distributed (IID) assumption is often made for training and test datasets to imply that all samples belong to the same distribution. However, to make the training process more efficient, the CSC algorithm training data may contain many more errors than usual, which will break the IID assumption and cause poor performance during inference.

To illustrate this phenomenon, we performed a detailed test of SpellGCN on *Triple Door*'s manuscript. *Triple Door* [27] is a famous novel, written by Hanhan. However, its originality has been widely questioned. After a renowned debate with Fang Zhouzi in 2012, Hanhan published *Triple Door*'s handwritten manuscript to prove his originality. There are still a number of spelling errors in that manuscript, which need proofreading before it can be published. Proofreading before the publication of novels or news articles is the most common application scenario of Chinese Spelling Error Correction. As the only published unedited original manuscript of a modern book to the best of our knowledge, we consider *Triple Door*'s manuscript as a great dataset to test the CSC algorithm's real application performance. First, we selected the first 10 pages of the manuscript, including 17 typos. Then, we divided the text into 110 sentences (error rate of 0.15), 16 of which contained errors. The average length of these sentences was 37.4. Next, we used SpellGCN to correct these sentences. We kept the hyperparameter settings of the model consistent with its official open-sourced implementation. During this experiment, we used Precision, Recall, F1 score and False Positive Rate (FPR) as the metrics.

We chose the model's result index on the SIGHAN15 test set (error rate of 0.5) as a comparison, containing 1100 sentences of which 550 sentences contained errors. As shown

in Table 1, when SpellGCN performed spelling error correction on *Triple Door*, the result index was far lower than the result obtained by the model on the SIGHAN15 test set, and the FPR value was closer to twice that of the latter. The uneven data distribution and the low error rate caused this phenomenon. Improving the performance of the algorithm model in actual error correction tasks will promote the application of such methods in the field of Chinese Spelling Error Correction.

Table 1. CRC results of SpellGCN on *Triple Door* at character level. P, R, F represent Precision, Recall, and F1 score. We also give the correction's False Positive Rate (FPR) at the sentence level. The FPR value represents the rate of false, corrected sentences to the total number of correct sentences.

	Detection			Correction			
	D-P	**D-R**	**D-F**	**C-P**	**C-R**	**C-F**	**FPR**
SIGHAN15	85.9	80.6	83.1	85.4	77.6	81.3	13.2
Triple Door	21.1	47.1	29	15.8	35.3	21.7	26.6

4. Approach

In this section, we first propose a method to solve the limitations of the current error correction algorithm, and then elaborate on the implementation process of our approach.

4.1. Post-Processing

We introduced a post-processing operation to help the algorithm avoid unnecessary replacements as much as possible in actual error correction tasks. Post-processing is a pervasive operation in various natural language processing tasks. Typically, in machine translation, the post-edit [28] corrects the model's first translation content, improving the final translation effect. In the past year, many abstract text summary algorithms have applied post-processing to prevent factual errors [29]. In the Chinese Spelling Error Correction field, further processing of the language model's correction results has gradually become a common method, such as FASPell [2] and adaptable filtering [24]. From the perspective of artificial intelligence, no matter how complex a neural network is, it is just a simple reflection model. So, the model's inference ability is always limited. We can improve the models' performance by adding planning and strategy. The post-processing operation is the first step from a simple reflection model to a more complex model.

As mentioned in Section 3, the core reason why SpellGCN and other SOTA models perform so poorly, in reality, is that the data in the training set and application scenarios do not conform to the assumption of independent and identical distribution. However, we cannot use data with the same distribution as the application scenario to train the model because the training data, which contain a few errors, cause the training process to be hugely inefficient, and the model cannot learn sufficient error-correction information. Thus, we can only separate model training and inference and use a post-processing operation to bridge the gap between the two.

Making models imitate humans as much as possible is one of the secrets to making machines smarter. The "two-step" error correction method is usually a standard behavior when humans perform Chinese Spelling Error Correction. In the first step, humans determine the candidate set for the suspected error in the sentence. The candidate set contains all possible modifications to the wrong character. In the second step, they analyze the connection between the previously obtained candidate character and the original character. Then, humans replace these two characters to judge whether the sentence's meaning has changed. They repeat the second step continuously to select the most appropriate candidate character as the correct one. The most current algorithms only complete the humans' first step in the error correction task and then decide the character with the highest probability value among the candidate sets as the error correction result.

Therefore, we added a post-processing operation after the existing error correction model. This operation is a classification process. Its input is the suggested error correction result given by the error correction model, and the output chooses whether to adopt this suggestion. If this classification model is accurate enough, it may slightly reduce the recall rate of text error correction while significantly improving its precision.

4.2. Implementation of Post-Processing

An extraordinary classification model should notice these points: the feature set and the classification model itself. We first elaborate on the features needed to post-process in the next section. After that, we introduce the chosen classifier and the output of our method.

4.2.1. Features Set

After analyzing the characteristics of Chinese Spelling Error Correction and human behavior when they perform these problems, we finally determined more than ten features, such as **sos** and **sop** for the classifier to screen the candidate character. Furthermore, we gave a schematic diagram of the replacement suggestion feature (水 → 苹, provided in Figure 1).

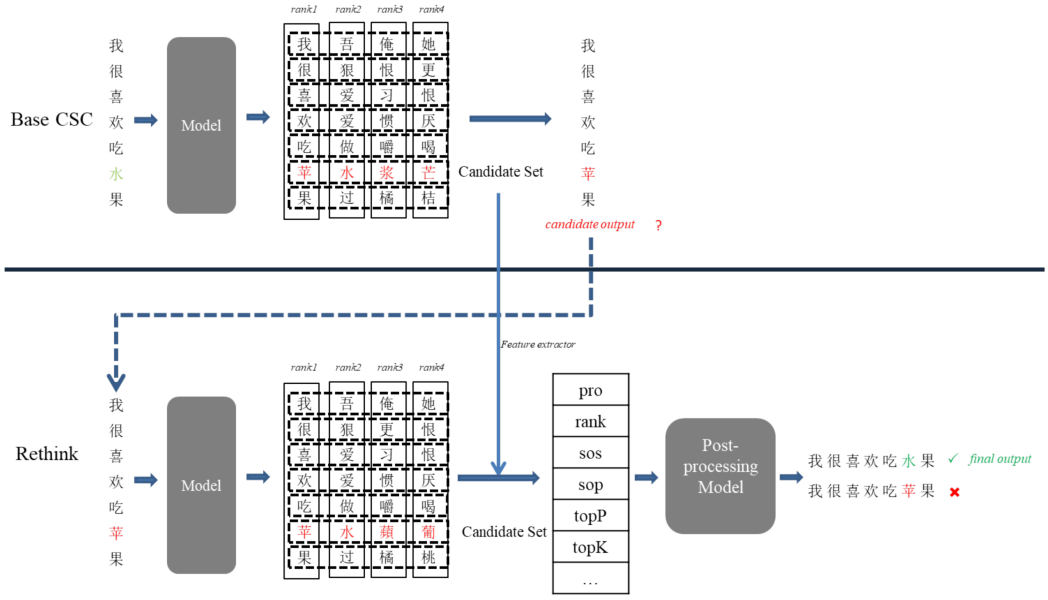

Figure 1. An example of introducing post-processing operation. The base model (the upper part of the figure) made a unnecessary change 水 to 苹 after correcting this sentence. However, for the PoP model (the lower part of the figure), we proposed avoided this change.

Probability and Rank

In the Chinese Spelling Error Correction task, the model gives a candidate set (1) that contains all possible characters in the vocabulary whose size is **V** for each original character in the sentence. Based on a given context, the model obtains each character's probability in the candidate set, appearing here through the softmax function. The sum of all characters' possibilities is 1. In theory, characters with higher probability are more likely to become target characters.

As we mentioned in Section 2, various BERT-based algorithms simply choose the character with the largest probability value as the correction output. We agreed that the character's probability should be one of the most important features, but it should not be the only one. Rank of probability does also matter. The different specific values of the probability affect the classification result. For example, there are two corrected candidate characters, which have the highest probability. Their corresponding possibilities

are $pro1 = 0.9$ and $pro2 = 0.3$, respectively. At this time, the classification model tends to judge the former ($pro1 = 0.9$) as correct, but the latter ($pro2 = 0.3$) as an error because $pro1 \gg pro2$ in this type of attribute. To make the model no longer affected by the different quantization values of probabilities, we ranked each character in the candidate set (1) according to the probability value from largest to smallest (rank = 1, 2, 3, 4, ...). In the previous example, the rank value corresponding to the two correct characters is $rank = 1$ because they have the highest probability, which satisfies the property that samples of the same category in the classification model have the same characteristics. Due to the Chinese text's errors providing helpful information for correction, we took the original character as a candidate character and look for the candidate set at that position to obtain its corresponding probability and rank, under the condition that the feature values of the candidate character are $canPro = 0.5$ and $canRank = 1$. While the original typo's features are $originalPro = 0.1$ and $originalRank = 89$, respectively, the model believes that this candidate character is a reasonable correction. While these two features are 0.48 and 2, the model must carefully consider whether the error correction is appropriate. We used the **pro** and **rank** features to represent the probability value and rank value of probability; Figure 2 shows an example of these two features' calculation.

$$C = [c_1, c_2, c_3, ...], \quad len(C) = V; \quad \sum_{i=1}^{V} p(c_i) = 1 \tag{1}$$

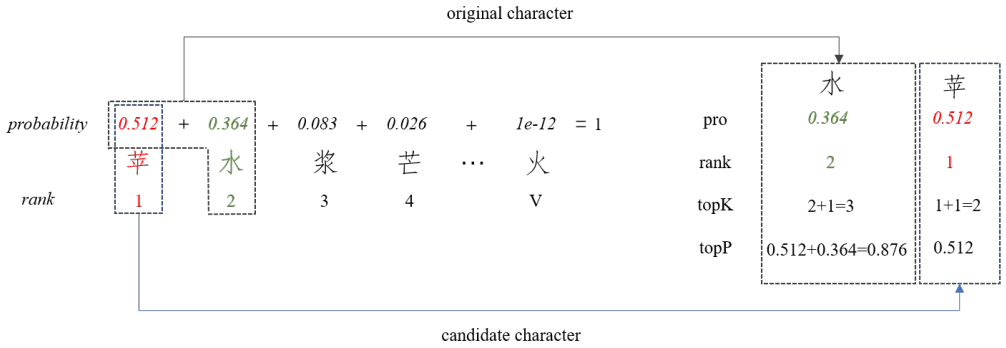

Figure 2. Across the candidate set, our approach obtains some features of 水 and 苹: **pro**, **rank**, **topK**, **topP**.

Phonetic and Morphological Similarity

Chinese characters are one of the oldest writing languages and have been developed and changed over the last three thousand years, from the original pictographic characters to today's simplified characters. Each character's shape and pronunciation are more or less related to the meaning of itself. There are similar pronunciations or forms between characters with related meanings. Observing the errors in Chinese text, we can find that most of the cases are that the error character and the correct character belong to the category of near-phonetic characters or near-form characters. Therefore, the degree of pronunciation and form proximity is an essential factor to judge whether the candidate character is reasonable. We introduced two features, **sos** and **sop**, to measure the similarity between the candidate character and original character in terms of shape and sound. To get a clear numerical expression, we built a Chinese character disassembly and pronunciation statistics file, and then calculated these two feature values by the heuristic method described in FASPell [2], as shown in Figure 3.

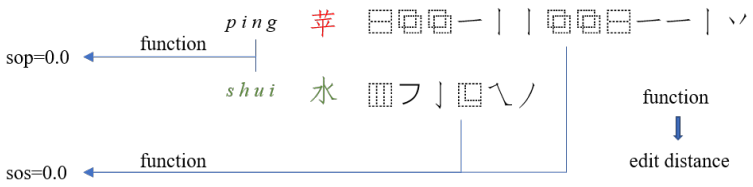

Figure 3. Our method calculates the phonetic similarity (**sop**) and glyph similarity (**sos**) between 水 and 苹 based on the edit distance.

Text Generation's Inspiration

Text generation is another hot research topic in natural language processing. Both text generation and spell checking try to output correct and high-quality text. Researchers directly used characters with the largest probability value in the model's last layer as the generation results in the early stage of text generation. This generation method leads to low diversity of the generated text. Therefore, researchers introduced sampling methods to improve diversity. However, sampling methods may lead to another problem: characters with very low probability will inevitably be sampled, which is extremely harmful to the generated text's overall fluency. Top-k sampling [30] and top-p sampling [31] are two improved sampling algorithms. They truncate low-probability words with accumulated rank or probability to improve the generated text's quality. Inspired by text generation tasks, we introduced the two attributes of **topP** and **topK** in order to help the final output results to get rid of the character's frequency. Following the specific implementation in the text generation task, we redefined these two attributes' calculation method. Firstly, we constructed a probability set (the uppercase English letter **P** represents this set in the formula) by replacing all characters with their corresponding probabilities in the candidate set (1). The original character's value of **rank** plus one is the value of **topK** since their definitions are very close. Then, we added up all the probability values higher than the character's probability value (the lowercase English letter p represents it in the formula), including itself as the value of topP, as shown in Figure 3.

$$topK = rank + 1 \tag{2}$$

$$topP = \sum p_i, \quad p_i \subset P, \quad p_i \geq p \tag{3}$$

Error Uncertainty

Due to the characteristics of the Chinese text itself, we found that changing some correct characters to many others can still ensure that the entire sentence is grammatically right but the original sentence's meaning is changed. When this happens, the original character should not be replaced, even if there are other reasonable candidates. Therefore, it is significantly inaccurate to determine that the error character relies on the character with the highest probability. We put forward an extreme hypothesis that if the probabilities of all characters appearing in a particular position of the sentence are the same, it means that every character is reasonable in that position. In this case, this position has the greatest uncertainty [32], and the error's possibility is the smallest. So, we introduced the feature of **uncertainty** to indicate the probability of error. We calculated this value as follows, as indicated in Figure 4.

$$uncertainty = -\sum p_i * \log p_i, \quad p_i \subset P \tag{4}$$

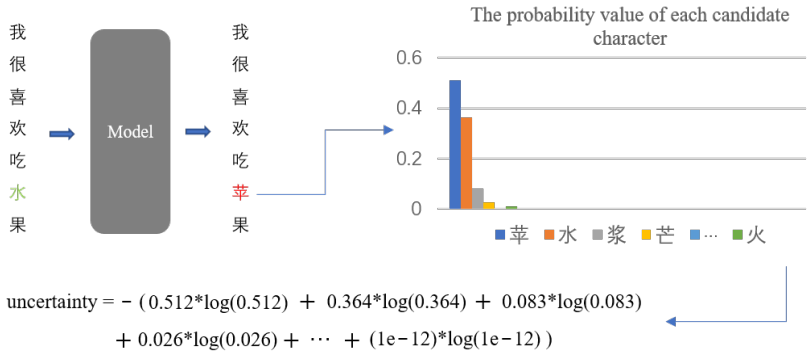

Figure 4. Our method calculates the character (水) position's error uncertainty **(uncertainty)** in the original sentence.

Sentence Similarity

A qualified correction will not change a sentence's meaning, or it should make the semantic change of the sentence as small as possible. Therefore, we introduced the attribute **similarity** to represent the semantic similarity of sentences before and after spelling correction. Fivez et al. [33] used a similar method to calculate a weighted cosine similarity between the vectorized representation of a candidate and the misspelling context. In our Chinese Spelling Error Correction system, we mapped each character in the sentence to a vector base on the embedding table. We used $\mathbf{X_i}$ and $\mathbf{Y_i}$ to represent the i-th character's vector in the input and output sentence, which contain N characters, respectively. Then, we took the cosine similarity between these two vectors as the each character pair's similarity value. Finally, we averaged all character pairs' cosine similarity as the semantic **similarity** between the input sentence and the output sentence, as shown in Figure 5.

$$similarity = \frac{1}{N}\sum_{i=1}^{N} cosine(\mathbf{X_i} * \mathbf{Y_i}) \tag{5}$$

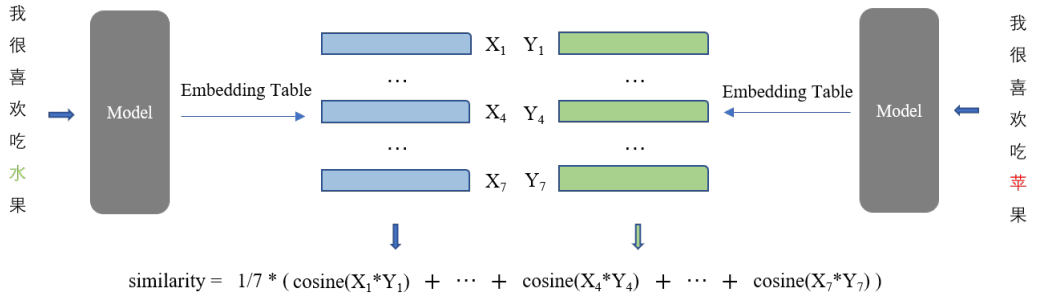

Figure 5. Our method calculates the semantic similarity **(similarity)** when the model changes 水 to 苹. This process need the cosine function and model's embedding table.

Replace and Re-Estimate

As mentioned in Section 4.1, humans usually follow two steps to solve the task of Chinese Spelling Error Correction. In the first step, they determine the candidates for the typo in the sentence. In the second step, they analyze whether each candidate character's replacement will change the sentence's semantics to select the most suitable candidate character. However, the current algorithm is analogous to humans' first step in solving error-correction tasks. Fitting to human behavior as closely as possible is the path for machines to become more intelligent. Therefore, combining with humans' second step to correct text errors is a crucial point. We used output candidate characters to replace the original sentence, and then took it as the model's new input to obtain the **pro**, **rank**, **topK**, **topP** repeatedly as shown in the lower part of the Figure 1.

4.2.2. Classifier

We relied on the classifier to determine whether the suggested replacement of the error correction model was reasonable. The candidate set is actually composed of correct replacement and incorrect replacement, so the classifier is actually a binary classifier. In specific implementation, the samples used to train the classifier have three characteristics: (a) the relationship between features and labels is a non-linear mapping; (b) the number of samples that can participate in training is relatively small; and (c) training samples have a lot of redundancy. So, we chose **SVM** [34] with a polynomial kernel as the classifier to filter out those unreasonable replacement suggestions. Before training the model, we divided the training samples into two categories (positive categories—correct replacement; negative categories—incorrect replacement) and ensured that the number of samples in the two types was basically the same.

4.2.3. Final Output

After inputting the sentence containing the error into the model, the model infers the target character at each position based on the input sentence's semantics to achieve the purpose of Chinese Spelling Error Correction. The activation function softmax is introduced into the last layer of the model to determine the target character's probability distribution in the vocabulary. The existing algorithms mostly select the character with the largest probability value from the candidate set (1) as the final output (6). When the candidate character with the highest probability is inconsistent with the original character, we also called this a replacement suggestion. We filtered the candidate set by setting a θ to achieve the simplest post-processing operation. The uppercase English letter **X** represents the original Chinese character in the formula. Only when the candidate character with the maximum probability value is greater than the given θ, will it be the correct output. Otherwise, we retained the initial character (7). We set the θ to 0.5 and use it as a baseline for post-processing operation.

Unlike the above two output results, our approach screens unnecessary replacement suggestions based on richer information. Firstly, we obtained all features between each original character and candidate character of the largest probability value, which is a replacement suggestion. Then, we relied on trained **SVM** to determine the suggestion's class. Only when this suggestion belongs to the positive class is it the final output. Otherwise, we kept the original character (8).

$$output = c_i, \quad max(p(c_i)) \tag{6}$$

$$output2 = \begin{cases} c_i, & \text{if } max(p(c_i)) \geq \Theta \\ X, & otherwise \end{cases} \tag{7}$$

$$output3 = \begin{cases} c_i, & max(p(c_i)), \quad c_i \subset Positive \\ X, & otherwise \end{cases} \tag{8}$$

4.3. An Example of Post-Processing

Figure 1 shows an example of our approach. The base model gives a replacement suggestion for a sentence that may contain errors. Although there is no problem with the revised sentence, it changes the semantics of the original sentence. So the suggestion by the model is not right, which should be dropped. In our approach, we obtain all the features introduced in the Section 4.2.2, use the trained classifier to filter this suggestion, and reject this wrong suggestion based on the judgment result. It effectively prevents the language model from modifying the original correct characters in the sentence.

5. Experiments

In this section, we introduce the details of our experiment. First, we describe all the experimental datasets, three baseline models, and evaluation metrics. After that, we not

only give the detailed experimental results, but also analyze and discuss the influence of different hyper-parameter settings in the experimental process.

5.1. Experiment Data

Training Data: The training data in the experiment come from the SIGHAN13-15 datasets, and we introduced an additional 271,000 pieces of data constructed through automatic generation [35]. The training data used to process the classifier come from a part of the SIGHAN2013 [5], SIGHAN2014 [25], and SIGHAN2015 [26] benchmarks.

Test Data: To evaluate each algorithm's performance, we used the remaining part of the official SIGHAN13-15.

In the experiment, we pre-processed all the data: (a) replace all the traditional characters in the sentence with simplified characters; and (b) filter out some sentences whose length exceeds 128. A detailed list of statistics is listed in Table 2 above.

Table 2. Statistics information of the used data resources. Train-PoP is the data used to train the classifier. It comes from the original three SIGHAN test sets. We selected 500 sentences from each test set to form the training set.

Training Data	Line	Avg. Length	Errors
Wang et al. [35]	271,329	42.5	381,962
SIGHAN13	350	49.3	350
SIGHAN14	6528	49.7	10,089
SIGHAN15	3174	29.0	4238
Total	281,381	42.6	386,639
Test Data	**Line**	**Avg. Length**	**Errors**
SIGHAN13	498	77.5	634
SIGHAN14	562	50.1	694
SIGHAN15	600	35.9	702
Total	1660	53.2	2030
Train-PoP	1500	48.4	1821

5.2. Baseline Models

We demonstrate our method by comparing three basic baseline models.

- **BERT** [1]: Taking character embedding technology as the softmax layer in model's top layer, we use the training data presented above to fine-tune this model.
- **SpellGCN** [4]: Through the introduction of graph coding in the upper layer of the model, the near-sounding and near-form information between Chinese characters are put into the model. We use the same settings as in the paper to train the model.
- **Filter by probability**: Based on the last layer of the model's output probability, we introduce a simple filtering operation by setting a threshold. We put this value to 0.5 in the experiment. When the probability of a candidate character is higher than this threshold, we take it as a valid candidate. Otherwise, we keep the original character (7).

5.3. Evaluation Metrics

In the field of Chinese Spelling Error Correction, we frequently use Precision, Recall, F1 score, and False Positive Rate (FPR) to evaluate the algorithm performance. The first three indicators include the character level and the sentence level. At the sentence level evaluation, only when all the errors in a sentence have been correctly detected or corrected can we accept this sentence as the right result. The FPR value represents the percentage of false corrected sentences' number in the correct sentences' total number.

5.4. Experiment Setting

For a fair comparison, all models in our experiment were based on the BERT-Base, that is, a 12-layer transformer with 768 hidden size, 12 attention heads and GLEU [36] activation. We fine-tuned the BERT and trained the SpellGCN, using AdamW [37] optimizer for six epochs with a batch size of 32 and a learning rate of 5×10^{-5}. For the base post-processing model, we set $\theta = 0.5$ to filter candidate characters whose probability is less than this value. In the implementation of our post-processing operation, we selected all features to train SVM, whose threshold is also the default value ($\theta = 0.5$).

5.5. Results

Table 3 presents that using the filter by probability cannot improve the results of the experiment. We observed the experimental results on SIGHAN13-15. There was no change in all of the model's index values after a base post-processing operation. This result also showed that a small part of the candidate set was filtered when we set the probability threshold $\theta = 0.5$. Therefore, we must design a detailed post-processing operation to get rid of the inaccuracy that only depends on probability.

Table 3. The test results after the post-processing operation on the two basic models. We recorded detailed different test indicators of two different subtasks: error detection and error correction on the test set in terms of character level and sentence level. P, R, F represent Precision, Recall, and F1 score. In the sentence level test, we also introduced the False Positive Rate (FPR). We intensified the improved results compared to the original model after introducing post-processing operation as an experimental comparison. FP denotes filter by probability, and PoP denotes post-processing.

	Character Level						**Sentence Level**							
	Detection			**Correction**			**Detection**				**Correction**			
SIAHAN13	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	FPR	C-P	C-R	C-F	FPR
BERT	83.3	**90.7**	86.9	81.6	88.8	85.0	77.0	73.1	75.0	(-)	74.8	71.1	72.9	(-)
SpellGCN	84.6	**89.6**	87.0	83.6	**88.5**	86.0	78.6	73.7	76.1	(-)	77.3	72.5	74.8	(-)
BERT-FP	83.7	90.5	87.0	82.1	**88.9**	85.3	77.4	73.5	75.4	(-)	75.3	71.5	73.3	(-)
SpellGCN-FP	84.8	89.1	86.9	83.9	88.2	86.0	78.4	73.1	75.7	(-)	77.4	72.1	74.6	(-)
BERT-PoP	**92.4**	82.8	**87.4**	**90.5**	81.1	**85.5**	**83.9**	**73.5**	**78.4**	(-)	**81.7**	**71.5**	**76.2**	(-)
SpellGCN-PoP	**91.7**	85.6	**88.6**	**90.5**	84.5	**87.4**	**84.4**	**76.1**	**80.0**	(-)	**83.1**	**74.9**	**78.8**	(-)
SIAHAN14	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	FPR	C-P	C-R	C-F	FPR
BERT	78.6	**79.3**	**78.9**	76.6	**77.4**	77.0	66.7	**69.8**	68.1	25.9	65.0	**68.0**	66.4	26.8
SpellGCN	79.4	**77.1**	**78.3**	77.9	**75.7**	76.8	65.8	68.3	67.0	26.2	64.7	67.3	66.0	26.1
Bert-FP	78.6	78.6	78.6	77.4	**77.4**	**77.4**	67.0	69.4	**68.2**	25.5	65.6	**68.0**	**66.8**	26.2
SpellGCN-FP	79.7	76.6	78.2	78.7	**75.7**	**77.2**	66.1	**68.7**	**67.3**	26.1	65.4	**68.0**	**66.7**	26.4
BERT-PoP	**86.3**	64.2	73.6	**83.7**	62.3	71.4	**68.1**	56.9	62.0	**21.1**	**66.0**	55.2	60.0	**22.2**
SpellGCN-PoP	**84.7**	67.4	75.1	**83.5**	66.4	74	**66.8**	60.1	63.3	**23.0**	**66.0**	59.4	62.5	**23.4**
SIAHAN15	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	FPR	C-P	C-R	C-F	FPR
BERT	78.6	**80.2**	**79.4**	74.6	**76.1**	75.4	67.3	**70.5**	**68.9**	25.5	64.4	**67.5**	65.9	27.2
SpellGCN	79.1	**80**	**79.4**	75.1	**75.6**	**75.3**	67.9	**72.6**	**70.2**	25.5	65.4	**70.0**	67.5	27.0
Bert-FP	79.1	79.7	**79.4**	75.6	**76.1**	**75.9**	67.1	70.0	68.5	25.5	64.8	**67.5**	**66.1**	26.8
SpellGCN-FP	79.0	79.2	79.1	75.2	75.4	**75.3**	67.8	72.3	70.0	25.5	65.6	69.9	**67.7**	26.8
BERT-PoP	**87.0**	70.0	77.3	**81.9**	65.5	72.8	**71.8**	63.7	67.5	**20.0**	**68.7**	61.0	64.6	**21.7**
SpellGCN-PoP	**84.8**	73.4	78.6	**80.1**	69.3	74.2	**71.2**	67.8	69.5	**21.5**	**68.0**	64.7	66.3	**23.4**

BERT and SpellGCN both had a very significant improvement in precision after the post-processing operation. For precision at the character level, these two models basically increased by 6% to 10%. However, the changes of the models' F1 scores were different on SIGHAN13-15. On the SIGHAN13 test set, all other indicators increased, except for the recall rate. Especially at the sentence level, the F1 score of detection and correction improved by about 5%. On the SIGHAN14 test set, the recall rate descended significantly, which led to the F1 score also descending by about 5%, whether at the character level or sentence level. On the SIGHAN15 test set, the effects of the post-processing operation

on precision and recall were balanced, so the F1 score was basically unchanged at the sentence level.

The post-processing operation can significantly reduce the model's False Positive Rate (FPR). FPR measures the unnecessary correction of the model, which is also the current algorithm's limitation in the actual error correction application. The FPR values of the two models descended by 3 % to 6%. This plays a significant role in reducing unnecessary model corrections.

6. Discussion

After introducing post-processing operations in the two models, the precision in all datasets was improved. Firstly, the basic model determines replacement candidates for each character in the sentence. Then, the post-processing operation combines the features and classifiers to filter out unnecessary corrections effectively. Because of this, the model's False Positive Rate is also significantly reduced.

Compared with the previous error correction algorithm, although the post-processing operation improves the model's precision, it comes at the cost of slightly reducing the recall because the post-processing filters out correct replacements when screening competitor replacements, which is disadvantageous for high recall tasks. However, in the Chinese Spelling Error Correction task, the corrections' preciseness is more critical than correcting all potential errors. Therefore, a model with higher precision rates is more suitable than one with a higher recall rate.

We also found that the performance of the post-processing operation on the three different test sets was quite different. For example, all indicators except recall were improved in SIGHAN2013, while the F1 score dropped on the other test sets. We believe that the uneven distribution of errors in the three datasets causes this phenomenon. As evidence, there are a large number of pronoun errors (他—he, 她—she) in the SIGHAN2014 test set. This type of error is challenging to identify for error correction models and post-processing operations.

Our approach is based on the observation that the distribution between the model's training text and its actual application text is inconsistent, hence existing algorithms produce many fault corrections. In our two-stage process of introducing post-processing operations, the basic model generates replacement suggestions in the first stage, and then the post-processing operation is taken as the second stage. One of the advantages of this approach is that without modifying a ready-to-use Chinese Spelling Error Correction model, it is possible to reduce incorrect error correction caused by the mismatch of the training text and the actual application text. The post-processing operation has a high generalization capability. Thus, in addition to BERT and SpellGCN, which we mentioned in this paper, researchers can select a greater variety of base models, according to different tasks.

7. Ablation Studies

In this part, we analyze the effect of several components on the classifier's performance, including selection of the different feature subsets and the setting of the **SVM**'s threshold. The ablation experiments were performed, using a test set, which contains 290 positive samples and the same number of negative samples. Then, we test the effects of post-processing operations on model performance under different error rates. We use the model's F1 value as the comparison index from the two subtasks: error detection and error correction.

Different Feature Subsets: Figure 6 shows the specific performance of the trained classifier on the test set when selecting different feature subsets. The related concept of text generation has a positive effect. Because the classification results used the probability-based feature, the subsets (pro and rank) are the same as the results when using the text generation features (topP and topK). Simultaneously, among all the classifiers trained on different subsets, these two classifiers have a better specific ability for negative samples. The classifier trained by using the uncertainty and the sentences similarity can recognize positive samples better than other sub-features classifiers. However, its ability to recognize

negative samples is the lowest. When we put all the features together, the resulting classifier achieves an ideal effect between positive samples and negative samples.

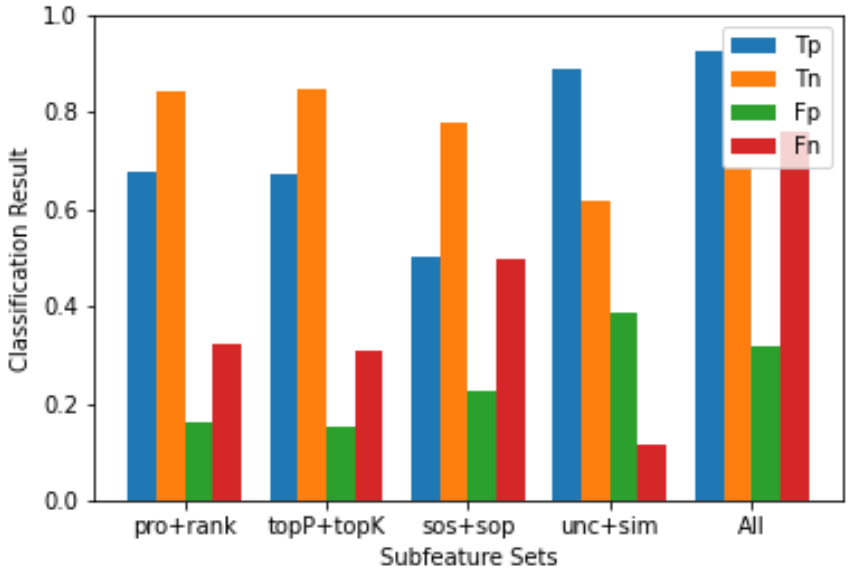

Figure 6. The different sets of features classifier on test set which contains 290 positive samples and 290 negative samples. The feature subsets used from left to right are **pro + rank**, **topP + topK**, **sos + sop**, **uncertainty + similarity (unc+sim)**, and **all features (All)**. The classification results of the classifier on the test set are marked with different colors.

Different error rate: At the end, we analyzed the effect of the post-processing operation under different error rates. In order to simulate real-life Chinese error texts, we used the official training set provided by the 2018NLPCC competition to construct them. The NLPPCC [38] is a Chinese misdiagnosis competition in which data have a similar distribution with the SIGHAN dataset. The no-error sentences from the competition's data are entirely correct. We randomly selected 3000 right sentences from the training set as our extra data. Table 4 gives the details of this data. We used BERT and post-processed BERT to correct the extra data and illustrate the results. Then we selected a part of these sentences and combined it with the test data to construct some more texts with different error rates (50%, 40%, 30%, 20%, and 10%). At last, we applied our BERT-PoP models as well as the baseline models to correct these texts and record the F1 score of the sentence level correction. The results are shown in Figure 7.

With the decrease in text error rate, higher model precision will result in an excellent F1 score. Figure 7 shows that when the error rate of the text is less than 20%, the BERT-PoP's F1 score is higher than that of BERT. The model's F1 score can increase by 10% or more when the error rate of the text is less than 10%. Therefore, the post-processing operation significantly promotes the model's performance when the text has a low error rate.

Different SVM's thresholds: The classifier's results in the test set, according to different thresholds, are shown in Table 5. It can be seen that the higher the threshold, the higher the precision. At the same time, it reduces the recall and accuracy of the classifier. So we should set the threshold according to the actual situation. When the situation requires a high precision rate, the classifier's threshold needs to be a higher value. In contrast, the classifier threshold needs to be lowered.

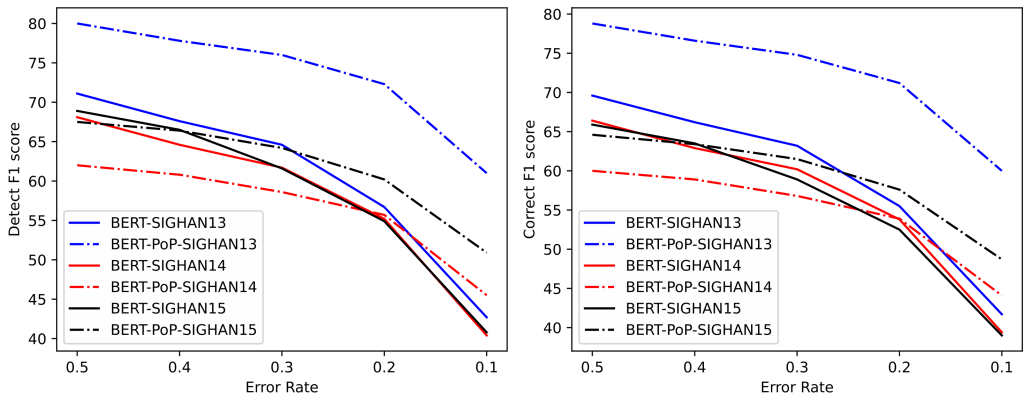

Figure 7. The F1 score of the model after the post-processing operation under different error rates (the percentage of error sentences in the text). Among them, the three colors represent different test sets. The figure's left side is the index results of error detection, and the right side is the index results of error correction. The straight line results from the original model and the dotted line results from adding the post-processing operation.

Table 4. Statistics information of the extra data. We selected 3000 sentences from the 2018NLPCC competition's training set.

Extra Data	Line	Avg. Length	Errors
NLPCC [38]	3000	18.8	0

Table 5. Different thresholds of SVM on test set. The default value of the threshold is 0.5.

Threshold	Accuracy	Precision	Recall
$\theta = 0.4$	80.8	85.3	91.3
$\theta = 0.5$	78.8	86.6	86.2
$\theta = 0.6$	75.5	87	80.9

8. Conclusions

In this paper, we put forward the limitations of applying existing Chinese Spelling Error Correction algorithms in practice. The model makes numerous unnecessary replacements that lead to a high False Positive Rate (FPR) value. In order to solve this problem, we proposed a method to perform a post-processing operation. This post-processing operation is based on many carefully choosed features, such as **sos**, **sop**, **topP**, **topK**, etc. These features can make full use of the pronunciation, shape, and semantics of Chinese characters. Unlike the traditional confusion set, we integrated these features directly into our model. More specifically, an SVM model is used to make the final correction decision. The post-processing operation can be broadly applied to various error correction models and significantly improve the model's performance in Chinese Spelling Error Correction applications, especially when the text contains only a few errors.

In future research, we will design more valuable features to prevent the classification model from filtering out correct corrections, which might ease the harm of the recall after post processing. In addition, we will investigate how to combine the characteristics of Chinese characters to study in depth the problem that the model cannot accurately correct pronoun errors.

Author Contributions: Conceptualization, W.G. and Z.C.; methodology, W.G. and Z.C.; data curation, W.G.; original draft preparation, W.G.; review and editing, Z.C. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
2. Hong, Y.; Yu, X.; He, N.; Liu, N.; Liu, J. FASPell: A Fast, Adaptable, Simple, Powerful Chinese Spell Checker Based on DAE-Decoder Paradigm. In Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019), Hong Kong, China, 4 November 2019.
3. Zhang, S.; Huang, H.; Liu, J.; Li, H. Spelling Error Correction with Soft-Masked BERT. *arXiv* **2020**, arXiv:2005.07421.
4. Cheng, X.; Xu, W.; Chen, K.; Jiang, S.; Qi, Y. SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check. *arXiv* **2020**, arXiv:2004.14166.
5. Wu, S.; Liu, C.; Lee, L. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing 2013, Nagoya, Japan, 14–18 October 2013.
6. Zhao, H.; Cai, D.; Xin, Y.; Wang, Y.; Jia, Z. A Hybrid Model for Chinese Spelling Check. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process.* **2017**, *16*, 1–22. [CrossRef]
7. Xin, Y.; Zhao, H.; Wang, Y.; Jia, Z. An improved graph model for Chinese spell checking. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014; pp. 157–166.
8. Chiu, H.W.; Wu, J.C.; Chang, J.S. Chinese spell checking based on noisy channel model. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014; pp. 202–209.
9. Jia, Z.; Wang, P.; Zhao, H. Graph model for Chinese spell checking. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 88–92.
10. Yang, S.; Zhao, H.; Wang, X.; Lu, B.l. Spell Checking for Chinese. In Proceedings of the International Conference on Language Resources and Evaluation, Istanbul, Turkey, 21–27 May 2012; pp. 730–736.
11. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.
12. Qiu, Z.; Qu, Y. A Two-stage Model for Chinese Grammatical Error Correction. *IEEE Access* **2019**, *7*, 146772–146777. [CrossRef]
13. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
14. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williamstown, MA, USA, 28 June–1 July 2001.
15. Duan, J.; Wang, B.; Tan, Z.; Wei, X.; Wang, H. Chinese Spelling Check via Bidirectional LSTM-CRF. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019.
16. Yao, Y.; Huang, Z. Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation. In *Neural Information Processing*; Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D., Eds.; Springer International Publishing: Cham, Swizterland, 2016; pp. 345–353.
17. Duan, J.; Yuan, Y.; Wang, H.; Wei, X.; Tan, Z. Research on Chinese Text Error Correction Based on Sequence Model. In Proceedings of the 2019 International Conference on Asian Language Processing (IALP), Shanghai, China, 15–17 November 2019.
18. Xie, H.; Li, A.; Li, Y.; Cheng, J.; Tang, Z. *Automatic Chinese Spelling Checking and Correction Based on Character-Based Pre-Trained Contextual Representations*; Springer: Cham, Switzerland, 2019.
19. Wang, H.; Wang, B.; Duan, J.; Zhang, J. Chinese Spelling Error Detection Using a Fusion Lattice LSTM. *arXiv* **2019**, arXiv:cs.CL/1911.10750.
20. Luong, M.T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-Based Neural Machine Translation. *arXiv* **2015**, arXiv:cs.CL/1508.04025.
21. Wang, Q.; Liu, M.; Zhang, W.; Guo, Y.; Li, T. *Automatic Proofreading in Chinese: Detect and Correct Spelling Errors in Character-Level with Deep Neural Networks*; Springer: Cham, Switzerland, 2019.
22. Han, Z.; Lv, C.; Wang, Q.; Fu, G. Chinese Spelling Check based on Sequence Labeling. In Proceedings of the 2019 International Conference on Asian Language Processing (IALP), Singapore, 4–6 December 2020.
23. Wang, D.; Tay, Y.; Zhong, L. Confusionset-guided Pointer Networks for Chinese Spelling Check. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July 2019.
24. Nguyen, M.; Ngo, G.H.; Chen, N.F. Adaptable Filtering Using Hierarchical Embeddings for Chinese Spell Check. *arXiv* **2020**, arXiv:cs.CL/2008.12281.
25. Yu, L.C.; Lee, L.H.; Tseng, Y.H.; Chen, H.H. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. In Proceedings of the Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014.

26. Tseng, Y.H.; Lee, L.H.; Chang, L.P.; Chen, H.H. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. In Proceedings of the 8th SIGHAN Workshop on Chinese Language Processing (SIGHAN'15), Beijing, China, 30–31 July 2015.
27. Han, H. *Triple Doors*; Writers Publishing House: Beijing, China, 2000.
28. Simard, M.; Foster, G. Pepr: Post-Edit Propagation Using Phrase-Based Statistical Machine Translation. In Proceedings of the XIV Machine Translation Summit, Nice, France, 2–6 September 2013; pp. 191–198.
29. Falke, T.; Ribeiro, L.F.; Utama, P.A.; Dagan, I.; Gurevych, I. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 29 July 2019; pp. 2214–2220.
30. Fan, A.; Lewis, M.; Dauphin, Y. Hierarchical Neural Story Generation. *arXiv* **2018**, arXiv:cs.CL/1805.04833.
31. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The curious case of neural text degeneration. *arXiv* **2019**, arXiv:1904.09751.
32. Anderson, J.R.; Michalski, R.S.; Mitchell, T.M. *Machine Learning: An Artificial Intelligence Approach*; Springer: Berlin/Heidelberg, Germany, 1984; Volume 2.
33. Fivez, P.; Šuster, S.; Daelemans, W. Unsupervised context-sensitive spelling correction of english and dutch clinical free-text with word and character n-gram embeddings. *arXiv* **2017**, arXiv:1710.07045.
34. Platt, J. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. In *Advances in Kernel Methods—Support Vector Learning*; MIT Press: Cambrige, MA, USA, 1998.
35. Wang, D.; Song, Y.; Li, J.; Han, J.; Zhang, H. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018.
36. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. Unpublished Manuscript. 2018. Available online: https://blog.openai.com/language-unsupervised/ (accessed on 22 June 2021).
37. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
38. Rao, G.; Gong, Q.; Zhang, B.; Xun, E. Overview of NLPTEA-2018 Share Task Chinese Grammatical Error Diagnosis. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, Melbourne, Australia, 17 July–4 August 2018.