Boosting Chinese ASR Error Correction with Dynamic Error Scaling Mechanism

Jiaxin Fan[1,2], Yong Zhang[1], Hanzhang Li[1,2], Jianzong Wang[1]*, Zhitao Li[1], Sheng Ouyang[1],
Ning Cheng[1], Jing Xiao[1]

[1]Ping An Technology (Shenzhen) Co., Ltd., China
[2]Lanzhou University, China

jzwang@188.com

Abstract

Chinese Automatic Speech Recognition (ASR) error correction presents significant challenges due to the Chinese language's unique features, including a large character set and borderless, morpheme-based structure. Current mainstream models often struggle with effectively utilizing word-level features and phonetic information. This paper introduces a novel approach that incorporates a dynamic error scaling mechanism to detect and correct phonetically erroneous text generated by ASR output. This mechanism operates by dynamically fusing word-level features and phonetic information, thereby enriching the model with additional semantic data. Furthermore, our method implements unique error reduction and amplification strategies to address the issues of matching wrong words caused by incorrect characters. Experimental results indicate substantial improvements in ASR error correction, demonstrating the effectiveness of our proposed method and yielding promising results on established datasets.

Index Terms: Chinese ASR error correction, dynamic error scaling mechanism, word-level feature fusion, phonetic information

1. Introduction

Automatic speech recognition (ASR) has become a widely used technology for transcribing spoken language into text. However, ASR systems are only sometimes accurate and often make errors in recognizing spoken Chinese. Text error correction techniques have emerged as effective means to address phonetic errors in the text output of ASR models[1, 2, 3, 4]. Several challenges make Chinese language error correction particularly difficult. Firstly, the Chinese character set is relatively large, with 3,500 commonly used characters and around 420 pronunciations[5], requiring more comprehensive error correction methods. Secondly, Chinese lacks clear word boundaries and only uses single words or characters, making it challenging to detect spelling errors and extract contextual semantics. Finally, Chinese words are morpheme-based[6] and typically consist of one to four characters, much shorter than English words. Errors in the phonetic similarity of the semantic center word in Chinese sentences can greatly impact overall semantic accuracy. This poses a challenge for recognizing phonological similarity errors with acoustic models, requiring language models to provide semantic aid.

Recent studies have tackled the problem of error correction in various ways. One popular approach is to treat it as a machine translation problem using a sequence-to-sequence (Seq2Seq)[7, 8] framework. In this approach, misspelled sentences generated by an automatic speech recognition (ASR)

system are taken as input, and a corrected sentence of the same length is generated as the output. Researchers have also used BERT-based[9] models for text correction, generating corrected characters for all input characters in parallel.

To improve the performance of contextual spelling correction, efforts have primarily focused on two issues. Firstly, researchers have aimed to enhance the quality of language models by introducing external word knowledge. For example, some studies propose incorporating phrase and entity knowledge[10] for error correction. Additionally, some studies[11] introduced part-of-speech (POS) features and semantic class features to enhance the performance of the model, and proposed an auxiliary task to predict the part-of-speech sequence of the target sentence. Other studies[12] have assisted the correction module by predicting the correct word segmentation boundaries from sentences containing misspellings and modifying the input of the embedding layer to incorporate word segmentation information. Secondly, researchers have explored leveraging phonetic information for text error correction. Some methods consider phonological similarities between pairs of characters, achieved through the increased decoding probability of characters with similar pronunciation or integrating such similarities into the encoding process via graph convolutional networks (GCNs[13]). Others utilization involves directly considering individual character pronunciation, specifically the pinyin. This approach encodes the pinyin of input characters to produce additional phonetic information[14] or decodes the pinyin of target correct characters to serve as an auxiliary prediction task[15, 16].

However, effective word-level feature fusion remains a challenge due to the risk of miscorrection caused by typos interfering with word segmentation results. Moreover, existing models need to more fully exploit the phonetic information of misspelled characters to ensure accurate predictions.

In this paper, we present a novel approach to text error correction based on a dynamic error scaling mechanism that overcomes the limitations of current methods. Our model offers two significant innovations that improve its performance.

Firstly, we explicitly model external lexical knowledge through targeted error reduction and amplification strategies. Unlike previous models that may overcorrect valid tokens or miss errors due to a lack of word-level information, our model prioritizes possibly wrong tokens during encoding and fixes potentially valid tokens during generation to avoid overcorrection.

Secondly, our model leverages word-level pinyin information to identify phonetic candidates instead of relying on character-based matching. However, character-based matching can lead to incorrect candidates, so we introduce a dynamic error scaling mechanism to more accurately identify related words. This approach allows us to explicitly leverage phonetic

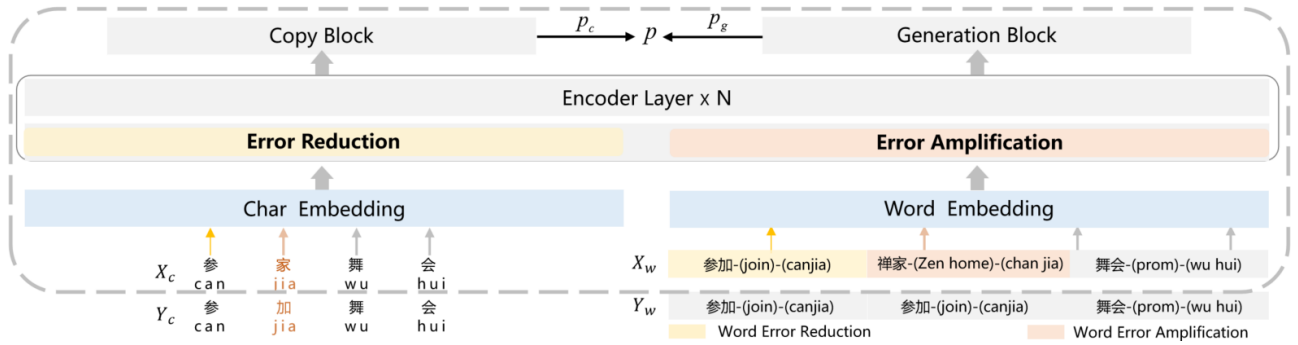*Corresponding author: Jianzong Wang, jzwang@188.com.

Figure 1: *Chinese ASR error correction model based on dynamic error scaling mechanism*

information and improve the accuracy of our model.

We put forth an efficient and straightforward text error correction approach in this paper. Our model, trained on the extensively used benchmark datasets, SIGHAN2013-2015[17, 18, 19], surpasses existing baselines when evaluated on the SIGHAN2015 test set, proving its competitiveness. One of the significant attributes of our approach is its simplicity compared to other sophisticated methods prevalent in the field. The effectiveness and robustness of our model are further exemplified through our comprehensive ablation study.

We highlight three major contributions to the field of text error correction in our work:

1. We present a unique dynamic error scaling mechanism adept at addressing word-level errors in erroneous texts.

2. We put forward a bi-directional 2-gram pinyin match that accurately harnesses the phonetic information of words, thus helping to identify plausible corrections.

3. Our model's superior performance over the existing baselines on the SIGHAN2015 test set demonstrates the effectiveness and competitiveness of our proposed methodology.

2. Method

In this section, we delineate our proposed method, providing its implementation details and model structure (refer to Figure 1). Our approach comprises a dynamic error scaling mechanism and a correction module, both contributing uniquely to the detection and rectification of errors.

The dynamic error scaling mechanism implements targeted error reduction and amplification strategies, while a bi-directional 2-gram pinyin match extracts the phonetic information of words. Our model processes both incorrect character sentences X_c and corresponding word-matching sequences X_w as input. After X_c is fed into the model, X_w is procured via the error scaling module, integrating a char-word attention mechanism to combine character and word-matching sequence information.

The correct versions, Y_c and Y_w, represent the targeted correction outcomes and ideal word-matching results, aiding in distinguishing between incorrect and correct word-matching sequences. In the correction module, we generate the correct result, employing a copy distribution simultaneously to avoid over-correction and improve accuracy.

In conclusion, our error correction model operates as follows: Given a character sequence $X_c = x_1^c, x_2^c, \ldots, x_n^c$ of length n, the model strives to produce a target sequence $Y_c = y_1^c, y_2^c, \ldots, y_n^c$ of equivalent length, where the typos in X_c are rectified. This task is posed as a conditional genera-
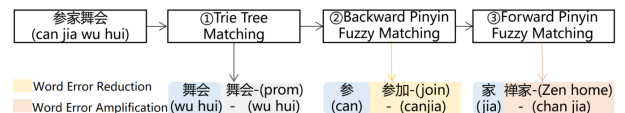

Figure 2: *Bi-direction 2-gram pinyin match flow chart of dynamic error scaling mechanism*

tion problem, aiming to maximize the conditional probability $P(Y_c \mid X_c)$.

2.1. Dynamic Error Scaling Mechanism

This section introduces a dynamic error scaling mechanism aimed at tackling inaccurate word matches resulting from character errors. In particular, the module integrates dictionary information into the model directly, extending character sequences into character-word pairs[20]. For enhancing the matching of incorrect words, we apply both error reduction and amplification strategies and integrate word-level phonetic information via bidirectional pinyin matching.

2.1.1. Bi-directional 2-gram Pinyin Matching

In our method, as illustrated in Figure 2, we utilize the dynamic error scaling mechanism to improve word matching accuracy. The process begins with trie tree matching (TTM) performed on all characters. For incorrect words resulting from incorrect characters, we incorporate a bidirectional 2-gram Pinyin match for word matching.

To accomplish this, we employ the PyPinyin library to obtain the Pinyin pronunciation of each character. Given that Chinese words commonly consist of 2-gram combinations, our algorithm pairs characters in incorrect words with corresponding 2-gram words from the dictionary in both backward and forward directions. It is important to note that the bidirectional 2-gram Pinyin matching algorithm considers 2-gram word matches in both the input text and the dictionary, rather than solely within the input text.

2.1.2. Error Reduction

In our error reduction strategy, we focus on two distinct scenarios to avoid overcorrection that may arise from integrating incorrect word information. Firstly, for those characters in words that are already correct within the sentence, we strive to accurately match them to the correct words as far as possible. This effort helps to preserve the integrity of the correct elements within the text.

Secondly, for correct characters found within incorrect words, we employ a specific set of strategies. The intention here is to ensure that these correct characters are not inappro-

priately matched to incorrect words due to the erroneous context they appear in. Notably, this strategy considers pronunciation errors in Chinese, such as flat and warped tongues, front and back nasal sounds, among others. For instance, we utilize the bi-directional 2-gram pinyin match to guide the character "参-(join)-(can)" found in an incorrect word to its correct word counterpart "参加-(join)-(can jia)". This way, we effectively mitigate the risk of overcorrection, thereby preserving the authenticity of the correct characters even within incorrect words.

2.1.3. Error Amplification

In the case of incorrect characters within erroneous words, it's imperative for the model to assimilate the incorrect word-level information to improve its capability in error detection and correction. For this reason, we adopt an error amplification strategy. During the bidirectional 2-gram pinyin match process, we limit the original characters to words with similar pronunciations and match incorrect words to incorrect characters. For example, the incorrect character "家-(home)-(jia)" can be matched to "禅家-(Zen home)-(chan jia)" for error amplification.

2.1.4. Dynamic Char-Word Attention

After we get the word sequence of all characters, we introduce a character-to-word dynamic attention[21] mechanism to determine the degree of correlation between each character and its matching. Specifically, for all words assigned to the i-th character, we denote their corresponding word vectors as $h_i^w = \{h_{i1}^w, \ldots, h_{im}^w\}$. Here, h_{ij}^w represents the vector of the j-th word assigned to the i-th character, where m is the total number of words assigned. The relevance of each word a_i can be calculated by (1), where W_{attn} is the weight matrix of bilinear attention, h_i^c is the i-th character vector. Through the application of nonlinear transformations to h_i^w, we can align different representations of character and words vectors. This approach enables our model to capture better and integrate contextual information of word-level features, thereby improving its text error correction ability.

$$a_i = \text{softmax}\left(h_i^c \mathbf{W}_{attn}\left(W_w h_i^w + b_w\right)^T\right) \quad (1)$$

Therefore, the weighted sum of all words can be obtained by:

$$z_i^w = \sum_{j=1}^{m} a_{ij}\left(W_j h_{ij}^w + b_j\right) \quad (2)$$

Finally, the weighted dictionary information is injected into the character vector via:

$$\tilde{h}_i = h_i^c + z_i^w \quad (3)$$

2.1.5. Differences From Confusion Sets

In distinguishing itself from traditional confusion sets, our dynamic error scaling mechanism introduces two innovative features. Firstly, our model uniquely utilizes a 2-gram pinyin match to explicitly leverage the phonetic information, thereby significantly enhancing its performance in correcting ASR output text. Secondly, we employ an innovative strategy of integrating word-level features dynamically. By incorporating a dictionary, our model successfully captures implicit syntactic and semantic knowledge, which markedly bolsters its error detection and correction capabilities.

2.2. Correction Module

Finally, we employ a correction module to correct erroneous sentences, which combines word-level feature and syntactic structure to correct errors, effectively improving the error correction accuracy. The final output of the correction module is a weighted sum of the generative distribution and the copy distribution.

2.2.1. Generative Distribution

The generative distribution, $\mathcal{P}_{gen} \in \mathrm{R}^v$ is computed by a single-layer feed-forward network with softmax normalization:

$$\mathcal{P}_{\text{gen}} = \text{softmax}\left(W_{\text{gen}}\tilde{h}_i + b_{\text{gen}}\right) \quad (4)$$

where $W_{gen} \in \mathrm{R}^{v \times 768}$ and $b_{gen} \in \mathrm{R}^{768}$ are generative parameters, v is the size of the vocabulary.

2.2.2. Copy Distribution

The copy distribution can directly copy the correct characters in the input sequence and only correct the generation distribution of the wrong characters, which is more suitable for the feature that most of the content of the input sequence in the text error correction task is correct, thus avoiding overcorrection. Denote the index of x_i^c in the vocabulary as $\text{id}(x_i^c)$, then the copy distribution of xi, $\mathcal{P}_{\text{copy}} \in \{0,1\}^v$, is a one-hot vector satisfying:

$$\mathcal{P}_{\text{copy}}[c] = \begin{cases} 0 & c \neq \text{idx}\left(x_i^c\right) \\ 1 & c = \text{idx}\left(x_i^c\right) \end{cases} \quad (5)$$

2.2.3. Output Probability

The final output of the correction module is a weighted sum of the generator distribution and the copy distribution, where the weights are the copy probabilities learned by the model. The copy probability ω is computed by a layer-normalized two-layer feed-forward network. The final output distribution $\mathcal{P}$ is calculated by:

$$\mathcal{P} = \omega \times \mathcal{P}_{\text{copy}} + (1 - \omega) \times \mathcal{P}_{\text{gen}} \quad (6)$$

Given a training sample (X, Y), the correction loss is defined as:

$$L = -\sum \log \mathcal{P}\left(Y_i \mid X_i\right) \quad (7)$$

where X_i is the i-th character of the wrong sentence, Y_i is the corrected character of X_i, and $\mathcal{P}$ is the output distribution.

3. Experiment

3.1. Datasets and Evaluation Metrics

The training data consisted of 10K manually annotated samples from SIGHAN [17, 18, 19] and 271K samples from Wang et al.'s work [22]. We evaluated our proposed model using the SIGHAN2015 [19] test dataset, which includes 550 positive samples and 550 negative samples. The negative samples represent text without any typos. Table 1 shows the data statistics. We compared the performance of our detection and correction model with several baseline methods [13, 15, 22, 23] using character-level[24] precision, recall, and F1 scores.

3.2. Baselines

To evaluate the performance of the proposed method, we compare it with the following baseline methods on the SIGHAN2015 test dataset:

Table 1: *Statistics of datasets*

Training Data	Sent	Errors
SIGHAN2013	350	343
SIGHAN2014	6,526	5,122
SIGHAN2015	3,174	3,037
Wang271k	271,329	381,962

Testing data	Sent	Errors
SIGHAN2013	1,000	1,224
SIGHAN2014	1,062	771
SIGHAN2015	1,100	703

- SoftMask [25] improves error detection in BERT by introducing a soft mask strategy.
- SpellGCN [13] combines the GCN network with BERT to model the relationship between characters.
- cBERT[15] has the same architecture as Bert, but is pretrained with knowledge of misspelled words.
- PLOME [15] proposes a pretrained model, cBERT[15], and incorporates phonological and visual features based on sequences of phonemes and strokes.
- CRASpell [24] constructs a noise modeling module based on cBERT, making their model robust against consecutive spelling errors. The model also includes a copy mechanism to handle over-correction.

3.3. Experimental Settings

Based primarily on the configuration put forth by Zhang et al.[25], we fix a maximum sentence length of 512, batch size at 32, and implement a learning rate of 5e-5. Our experimental setup incorporates encoders such as BERT[9] and cBERT[15], while employing the pre-trained word embedding provided by Song et al.[26].

3.4. Main Results

Table 2 displays the character-level performance of our cBERT-based model and its baseline methods on the SIGHAN2015 test set. The results demonstrate that our model achieves comparable character correction scores to CRASpell, while exhibiting higher recall and f-score in character detection.

Table 2: *The character-level performance on the SIGHAN2015 test set*

Method	Detection-level			Correction-level		
	P	R	F	P	R	F
SoftMask	75.5	84.1	79.6	96.7	81.4	88.4
SpellGCN	77.7	85.6	81.4	96.9	82.9	89.4
cBERT	83.0	87.8	85.3	96.0	83.9	89.5
PLOME	**85.2**	86.8	86.0	**97.2**	85.0	90.7
CRASpell	83.5	89.2	86.3	97.1	**86.6**	**91.5**
ours (cBERT)	83.6	**89.8**	**86.6**	96.2	86.3	91.0

3.5. Ablation Study

We conducted ablation studies to investigate the contribution of each component in our BERT-based model. The components examined were the copy mechanism, raw trie tree matching (TTM), and the dynamic error scaling mechanism (DESM).

Table 3 reveals that the integration of TTM into BERT improved all metrics over raw BERT and BERT with the copy mechanism. Adding bi-directional 2-gram pinyin match to TTM, forming the DESM, further boosted all scores except cor-

rection precision, which decreased compared to TTM. This indicates our model may over-correct errors for a better overall performance.

Table 3: *Ablation Study on SIGHAN2015: character-level*

Method	Detection-level			Correction-level		
	P	R	F	P	R	F
BERT	75.8	85.5	80.4	94.7	80.9	87.3
BERT+copy	78.1	85.8	81.8	95.7	82.1	88.4
BERT+TTM+copy	80.0	86.5	83.1	**96.2**	83.2	89.2
BERT+DESM+copy	**80.7**	**87.6**	**84.0**	95.6	**83.8**	**89.3**

Note: BERT and BERT+copy data from CRASpell[24].

3.6. Error Analysis

To analyze prediction errors, we conducted a statistical analysis of the Part-Of-Speech of error words in the SIGHAN2015 test set. Table 4 presents the original error distribution and the distribution after correction by our method based on BERT.

Table 4: *Comparison of Part-Of-Speech Distributions between Incorrect and Corrected Text Using Our Method*

	verb	noun	adv	pron	aux	others
Source data	32.1%	26.7%	10.4%	8.1%	6.3%	16.4%
BERT	3.4%	2.1%	0.2%	2.3%	1.5%	3.2%
ours (cBERT)	1.9%	1.5%	0.3%	2.2%	1.7%	1.4%

We use the words matched by the error scaling mechanism and the Jieba library to judge the Part-Of-Speech of the wrong characters. From a statistical analysis of the distribution of Part-Of-Speech tags for wrong words in the incorrect text, it can be seen that nouns and verbs occur more frequently, which mainly consist of 2-gram words. Our model with a dynamic error scaling mechanism can significantly correct 2-gram word errors, thus reducing this type of error significantly. However, errors associated with single characters, such as auxiliary words (e.g., 啊 (ah) and personal pronouns (e.g., 我 (I) and 你 (you)), often occur at the beginning or end of a sentence, making them challenging to correct. Additionally, our model performs poorly when encountering successive errors at the end of a sentence.

4. Conclusions

In this work, we've introduced a novel dynamic error scaling mechanism aimed at improving Chinese ASR by detecting and correcting phonetically erroneous outputs. This mechanism dynamically combines word-level features and phonetic information, effectively utilizing a bi-directional 2-gram pinyin match to leverage phonetics. Our strategy employs error reduction and amplification to enhance error handling. Benchmark experiments demonstrate our model's improved F1 performance over existing baselines, attesting to the effectiveness and superiority of our approach in Chinese ASR error correction.

5. Acknowledgement

6. References

[1] C. Anantaram, A. Sangroya, M. Rawat, and A. Chhabra, "Repairing asr output by artificial development and ontology based learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5799–5801.

[2] A. Mani, S. Palaskar, N. V. Meripo, S. Konam, and F. Metze, "Asr error correction and domain adaptation using machine translation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6344–6348.

[3] Y. Leng, X. Tan, L. Zhu, J. Xu, R. Luo, L. Liu, T. Qin, X. Li, E. Lin, and T.-Y. Liu, "Fastcorrect: Fast error correction with edit alignment for automatic speech recognition," in *Advances in Neural Information Processing Systems*, 2021, pp. 21 708–21 719.

[4] Z. Fang, R. Zhang, Z. He, H. Wu, and Y. Cao, "Non-autoregressive chinese asr error correction with phonological training," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 5907–5917.

[5] J. Duan, L. Pan, H. Wang, M. Zhang, and M. Wu, "Automatically build corpora for chinese spelling check based on the input method," in *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*. Springer, 2019, pp. 471–485.

[6] Z. Bao, C. Li, and R. Wang, "Chunk-based chinese spelling check with global optimization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 2031–2040.

[7] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5651–5655.

[8] O. Hrinchuk, M. Popova, and B. Ginsburg, "Correction of automatic speech recognition with transformer sequence-to-sequence model," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7074–7078.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Volume 1: Long Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[10] R. Zhang, C. Pang, C. Zhang, S. Wang, Z. He, Y. Sun, H. Wu, and H. Wang, "Correcting chinese spelling errors with phonetic pretraining," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP, (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021, pp. 2250–2261.

[11] X. Wu and Y. Wu, "From spelling to grammar: A new framework for chinese grammatical error correction," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 889–902.

[12] F. Li, Y. Shan, J. Duan, X. Mao, and M. Huang, "Wspeller: Robust word segmentation for enhancing chinese spelling check," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 1179–1188.

[13] X. Cheng, W. Xu, K. Chen, S. Jiang, F. Wang, T. Wang, W. Chu, and Y. Qi, "Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 871–881.

[14] H.-D. Xu, Z. Li, Q. Zhou, C. Li, Z. Wang, Y. Cao, H. Huang, and X.-L. Mao, "Read, listen, and see: Leveraging multimodal information helps chinese spell checking," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP, (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021, pp. 716–728.

[15] S. Liu, T. Yang, T. Yue, F. Zhang, and D. Wang, "Plome: Pre-training with misspelled knowledge for chinese spelling correction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, (Volume 1: Long Papers)*, 2021, pp. 2991–3000.

[16] T. Ji, H. Yan, and X. Qiu, "Spellbert: A lightweight pretrained model for chinese spelling check," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3544–3551.

[17] S.-H. Wu, C.-L. Liu, and L.-H. Lee, "Chinese spelling check evaluation at sighan bake-off 2013," in *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 2013, pp. 35–42.

[18] L.-C. Yu, L.-H. Lee, Y.-H. Tseng, and H.-H. Chen, "Overview of sighan 2014 bake-off for chinese spelling check," in *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*. Association for Computational Linguistics, 2014, pp. 126–132.

[19] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to sighan 2015 bake-off for chinese spelling check," in *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*. Association for Computational Linguistics, 2015, pp. 32–37.

[20] W. Liu, X. Fu, Y. Zhang, and W. Xiao, "Lexicon enhanced chinese sequence labeling using bert adapter," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL-IJCNLP, (Volume 1: Long Papers)*, 2021, pp. 5847–5858.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[22] D. Wang, Y. Song, J. Li, J. Han, and H. Zhang, "A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018, pp. 2517–2527.

[23] D. Wang, Y. Tay, and L. Zhong, "Confusionset-guided pointer networks for chinese spelling check," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5780–5785.

[24] S. Liu, S. Song, T. Yue, T. Yang, H. Cai, T. Yu, and S. Sun, "Craspell: A contextual typo robust approach to improve chinese spelling correction," in *Findings of the Association for Computational Linguistics: ACL*, 2022, pp. 3008–3018.

[25] S. Zhang, H. Huang, J. Liu, and H. Li, "Spelling error correction with soft-masked bert," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 882–890.

[26] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Volume 2 : Short Papers)*. Association for Computational Linguistics, 2018, pp. 175–180.