SpellBERT: A Lightweight Pretrained Model for Chinese Spelling Check

Tuo Ji, Hang Yan, Xipeng Qiu*
School of Computer Science, Fudan University
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
{tji19, hyan19, xpqiu}@fudan.edu.cn

Abstract

Chinese Spelling Check (CSC) is to detect and correct Chinese spelling errors. Many models utilize a predefined confusion set to learn a mapping between correct characters and its visually similar or phonetically similar misuses but the mapping may be out-of-domain. To that end, we propose SpellBERT, a pretrained model with graph-based extra features and independent on confusion set. To explicitly capture the two erroneous patterns, we employ a graph neural network to introduce radical and pinyin information as visual and phonetic features. For better fusing these features with character representations, we devise masked language model alike pre-training tasks. With this feature-rich pre-training, SpellBERT with only half size of BERT can show competitive performance and make a state-of-the-art result on the OCR dataset where most of the errors are not covered by the existing confusion set

1 Introduction

Spelling Check is to detect and correct Chinese spelling errors in sentences. However, it is a non-trivial task for Chinese spelling check because of the nature of ideographic language. Chinese has a large vocabulary including at least 3,500 common characters which leads to huge search space and an unbalanced distribution of errors.

Though hard to cover most of the misuses, their patterns could be roughly reduced to visual or phonetic errors (Chang, 1995) as shown in Figure 1. The former type of errors have similar shapes as correct ones and they are often caused by optical character recognition (OCR) or morphology-based input method. The other type of errors have similar pronunciation as original ones and they are usually caused by automatic speech recognition (ASR) or phonetic-based input method.

Previous work (Hsieh et al., 2013; Yu and Li, 2014; Wang et al., 2019a; Cheng et al., 2020) tend to employ a predefined confusion set to find and filter correction candidates. Confusion set is constructed by incorrect stats (Liu et al., 2010) and it has a mapping between visually similar pairs and phonetically similar pairs in accord with erroneous patterns. However, these models only learn a shallow mapping from confusion set and their performance is heavily dependent on the quality of confusion set. But it is hard to find an up-to-date and in-domain confusion set.

Figure 1: The two erroneous patterns. (a) is a phonetic error and its pinyin have overlap with the correct ones. (b) is a visual error and its radicals also have overlap

In this paper, we devise two pre-training tasks to model the two aforementioned erroneous patterns explicitly. To model visual errors, we introduce radical features. Chinese characters can be decomposed into various components namely radical. As for phonetic errors, we employ pinyin as features which are descriptions of pronunciation. We fuse these visual and phonetic features with character representations by relational graph convolutional network (Schlichtkrull et al., 2018). Likewise masked language model in BERT (Devlin et al., 2019), we randomly replace some characters and then predict the original visual and phonetic features with false input. Our model, SpellBERT, can intrinsically learn to correct errors based on visual or phonetic patterns rather than simple mapping. On the OCR dataset, where only a few errors are covered by confusion set, we make a state-of-the-art result and this indicates that SpellBERT can generalize well without depending on confusion set.

On resource-constrained scenarios for deployment, making a model lightweight is necessary. SpellBERT only has half size of BERT and is more efficient for these scenarios.

*Corresponding Author.

In summary, SpellBERT is independent on confusion set in training and inference phase. With only half size of BERT, SpellBERT can show competitive performance and generalize well.

2 Related Work

Current methods consider CSC as sequence generation problem or sequence labeling problem. Wang et al. (2019b) introduce copy mechanism to generate corrected sequence. Bao et al. (2020) unify single-character and multi-character correction by a chunk-based generative model.

Pretrained models (PTMs) have made a success on sequence labeling tasks (Qiu et al., 2020). Masked language model (MLM) is introduced as pre-training task to predict masked or replaced words conditioned on context. The mode of MLM is intuitively appropriate to be transformed to predict spelling errors and correct them. Significant progress has been made by power of PTM (Hong et al., 2019). Based on MLM, confusion set is applied to narrow search space for predicting correct characters. Cheng et al. (2020) constructed a graph by confusion set to help final prediction. Nguyen et al. (2020) raised an adaptable confusion set but its training process is not end-to-end.

Ideally, CSC corpus can be infinitely constructed by replacing words based on confusion set. Wang et al. (2018) generated 270k data by OCR- and ASR-based approaches. Zhang et al. (2020) created 5 million augmented data and Li et al. (2021) created 9 million augmented data by substitution-based method. Zhang et al. (2021) corrupted input sentence by randomly replacing characters with noisy-pinyin and the new pre-training task fitted well for CSC.

More recently, some methods also utilized phonetic and visual features in CSC. Liu et al. (2021) employed a GRU(Bahdanau et al., 2014) to encode pinyin sequence and Chinese strokes sequence as extra features. Xu et al. (2021) had similar design but they encoded pictures of characters to get visual features. Huang et al. (2021) enriched character representations by knowledge of audio and visual modalities. Our method is different from all of these work. For phonetic features, we regard pinyin as a whole but not a sequence. For visual features, we used radicals which are higher-level features than strokes. And we incorporate these extra features by graph neural network.

3 Approach

We treat CSC as a sequence labeling problem. An input sequence with n characters is represented as $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$. Our goal is to transform it into a target sequence $\mathbf{Y} = \{y_1, y_2, \cdots, y_n\}$. During which, incorrect characters will be detected and corrected. Obviously, the input and output share the same vocabulary and most of the output characters can be directly copied from input. The framework of our model is shown in Figure 2. It contains three parts, i.e., a BERT-based encoder, a feature-fusing module and a component for pretraining. We will progressively elaborate our design in detail.

3.1 An MLM-based Backbone

Many attribute the success of BERT (Devlin et al., 2019) to its MLM pre-training task. BERT randomly masked or replaced some tokens and then predict the original tokens. Regarding the masked and replaced tokens as spelling errors, BERT is properly adapted to be a spelling checker. Each input character x_i is indexed to its embedding representation $\mathbf{e}_i$ by the BERT-embedding-layer. Then $\mathbf{e}_i$ will be passed to BERT-encoder-layers to get a representation $\mathbf{h}_i$ as follows:

$$\mathbf{e}_i = \text{BERTEmbedding}(\mathbf{x}_i), \quad (1)$$

$$\mathbf{h}_i = \text{BERTEncoder}(\mathbf{e}_i), \quad (2)$$

where $\mathbf{e}_i, \mathbf{h}_i \in \mathbb{R}^{1\times d}$ and d is the hidden dimension. After that, the $\mathbf{h}_i$ will be computed similarities with all character embeddings to get a predicted distribution $\hat{\mathbf{y}}_\mathbf{i}$ over vocabulary as follows:

$$\hat{\mathbf{y}}_\mathbf{i} = \text{Softmax}(\mathbf{h}_i\mathbf{E}^T), \quad (3)$$

where $\mathbf{E} \in \mathbb{R}^{V\times d}$; $\hat{\mathbf{y}}_\mathbf{i} \in \mathbb{R}^{1\times V}$ and V is the vocabulary size. Here $\mathbf{E}$ refers to the BERT-embedding-layer and the i_{th} row of $\mathbf{E}$ corresponds to $\mathbf{e}_i$ in Equation 1. Finally we use the character x_k as the correction result for x_i whose $\mathbf{e}_k$ has the highest similarity with $\mathbf{h}_i$.

3.2 Fusing Visual and Phonetic Features

The above backbone lacks special modeling for this task. Chinese spelling errors can be roughly classified into two patterns. Visual errors have similar shapes as correct characters while phonetic errors have similar pronunciation. Some work utilize an external confusion set that has predefined mappings between visually similar pairs and phonetically similar pairs (Yu and Li, 2014; Wang et al.,

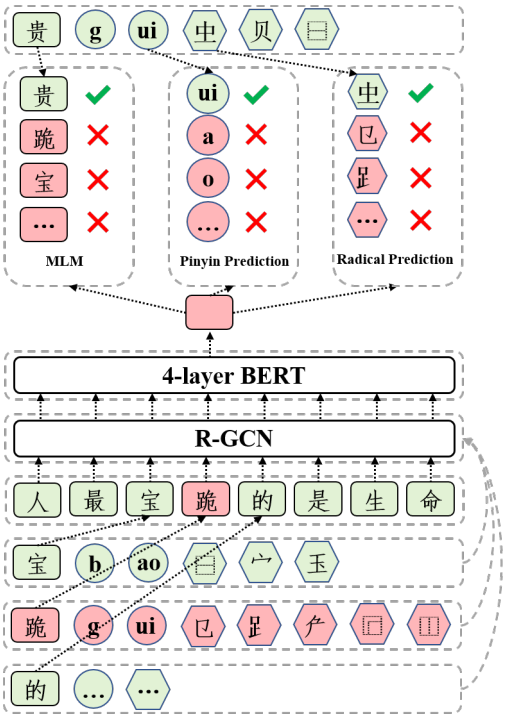

Figure 2: The architecture of SpellBERT. Green denotes correct characters while red denotes errors. Pinyin and radical features are fused by graph and then passed to a 4-layer BERT. We expect the model to do MLM, pinyin prediction and radical prediction of correct characters with false input.

2019a; Cheng et al., 2020). These models relied on confusion set to filter candidates but the confusion set might be out-of-date or out-of-domain.

To model the two erroneous patterns, we infuse character representations $\mathbf{e}_i$ with visual and phonetic features by incorporating radical and pinyin information. Chinese characters can be decomposed into components namely radicals and visual errors often have overlap radicals with the correct character. Pinyin is a sequence of pronunciation descriptions for Chinese characters and phonetic errors often have overlap pinyin. Based on the extra features, our model can automatically learn visually similar and phonetically similar mappings.

We employ a relational graph convolutional network (Schlichtkrull et al., 2018) short as R-GCN to infill multiple types of features into character representations $\mathbf{e}_i$ in Equation 1. We view characters as nodes and input sequence $\mathbf{X}$ can be organized as a line graph naturally. Both radicals and pinyin are viewed as nodes of graph as well. If a radical or pinyin belong to a certain character, we construct connections between them as edges. We regard these connections as different depending on the pair of nodes between them. Besides, we construct edges between neighboring characters because local context information is beneficial for better-incorporating pinyin and radical features. As a result, We define the following types of edges:

- An edge between a character and a radical
- An edge between a character and a pinyin
- An edge between a character and a neighboring character within a fixed-length context
- An edge between a character and itself

We initialize feature of character-node by character-embedding $\mathbf{e}_i$ in Euqation 1. To represent and update features of radical-node and pinyin-node, we also construct an extra embedding table which is initialized by averaging their most related character-embeddings. As shown in Figure 2, these features diffuse on a relational graph as following:

$$\hat{\mathbf{e}}_i = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{W}_r\mathbf{e}_j + \mathbf{W}_0\mathbf{e}_i\right), \quad (4)$$

where $\mathbf{e}_i$ means character-embedding of x_i and $\mathbf{e}_j$ means feature of connected node j; r denotes the type of edge; $\mathcal{N}_i^r$ refers to the set of connected nodes for edge type r; $\mathbf{W}_r$ is the transformation layer of edge type r and $\mathbf{c}_{i,r}$ is a problem-specific normalization constant which is set as $|\mathcal{N}_i^r|$ here. The final $\hat{\mathbf{e}}_i$ can be viewed as character representation enhanced by radical and pinyin information. Finally, we combine enhanced representation and original character-embedding and Equation 3 can be updated as following:

$$\mathbf{h}_i = \text{BERTEncoder}(\mathbf{e}_i + \hat{\mathbf{e}}_i), \quad (5)$$

where $\mathbf{h}_i$ denote the final representation of each character.

3.3 Enhanced Pretraining Tasks for CSC

It has been shown that external information can be better integrated into BERT by pre-training alike tasks (Peters et al., 2019; Zhang et al., 2019; Sun et al., 2020; Ma et al., 2020). Considering the radical and pinyin features are externally added by design, we devise two more pre-training alike tasks which are radical prediction and pinyin prediction.

In MLM, Devlin et al. (2019) randomly masked a percentage of input tokens and then predict these tokens. In radical and pinyin prediction, we randomly mask connections from characters to their radicals and pinyin and then predict the masked

connections. Through reconstructing connections, the model can learn a better representation that contains not only contextual information but also visual and phonetic information.

Same as MLM, we randomly choose 15% of characters to process. If a character is chosen, our potential practices are shown below:

- Keep it unchanged 10% of the time. Then predict the character itself, its radicals, and its pinyin. This is to match downstream fine-tuning where each character can directly see all of its radicals and pinyin.
- Replace it with [MASK] 60% of the time and mask all of its connections with a probability of 80%. Then predict the masked character and the masked connections.
- Replace it with a confusing word sampled from confusion set 30% of the time and mask all of its connections with a probability of 80%. Then predict the original character and its connections. This is to force our model to correct characters based on false radicals and pinyin of errors. Note that we only use confusion set in this stage to construct misspellings.

In our graph, edges have no representations and the graph is utilized only between BERT-embedding-layer and BERT-encoder-layers. So we transform the task of edge-prediction into token-classification. For each character x_i, we take one of its pinyin and radicals as ground-truth and negatively sample other pinyin and radicals that do not belong to the character. We use feature-embeddings of these pinyin and radicals as a classified layer to compute their similarities with $\mathbf{h_i}$ from BERT-encoder-layer in Equation 2. Related embeddings will be drawn close to each other, and unrelated embeddings will be pulled away from each other.

3.4 Reducing Parameters

Given the need of computational efficiency for deployment, it is necessary to get a lightweight model. We only use 4 layers of BERT to initialize, pre-train, and fine-tune our model and which reduces the total number of parameters from 110M to 55M. We also measure the inference speed of our lightweight model and the experiments result show that it has better time-efficiency compared with a 12-layer BERT.

4 Experiments

4.1 Pre-training Setup

We use BERT (Devlin et al., 2019) base as initialization and only the first 4 layers are utilized. Our model is implemented by PyTorch (Paszke et al., 2019) and DGL (Wang et al., 2019c). We randomly select 1M sentences provided by Xu (2019) as pre-training corpus and pad the sentences to a max length of 128. We set the learning rate as 5e-5, batch size as 1024, and pre-train 10K steps on 4 RTX 3090 for around 2 days.

4.2 Dataset and Fine-tuning Setup

We conduct CSC experiments on three widely used datasets SIGHAN14 (Yu et al., 2014), SIGHAN15 (Tseng et al., 2015), OCR (Hong et al., 2019) and mark them as csc_{14}, csc_{15} and ocr.

The original corpus of csc_{14} and csc_{15} was collected from essays written by learners of Chinese as a foreign language and it was in Traditional Chinese. Wang et al. (2019a), Zhang et al. (2020), and Nguyen et al. (2020) transformed it into Simplified Chinese and used augmented data provided by Wang et al. (2018). Because our pre-training corpus was in Simplified Chinese, we use the latter setting. We directly use the corpus provided by Cheng et al. (2020). Under this setting, the training set of csc_{14}, csc_{15} and the augmented data provided by Wang et al. (2018) are combined as a new training set. We fine-tune our model on the test set of csc_{14} and csc_{15} separately.

ocr is a Simplified Chinese dataset of which the sentences are much shorter and extracted from the entertainment domain. We only use the data from ocr to train and test and it has 4575 sentences in total.

For different datasets, we find the following ranges of hyperparameters work well: the batch size is set to among {32, 64}, the learning rate is set to among {1e-5, 2e-5, 3e-5} and the number of epochs is ranging from 5 to 20.

On csc_{14} and csc_{15}, we evaluate our model in sentence-level by the official tool (Tseng et al., 2015). And on ocr, the metric is in edit-level by a different official tool (Wu et al., 2013).

4.3 Results and Analysis

Main Results As shown in Table 1, we compare SpellBERT with recent work and a 4-layer BERT baseline. All of them are BERT-based which means that their number of parameters are at least twice

Model	Detection Level									Correction Level								
	ocr			csc_{14}			csc_{15}			ocr			csc_{14}			csc_{15}		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Nguyen et al. (2020) (BERT 12 layers)	-	-	-	82.5	61.6	70.5	84.5	71.8	77.6	-	-	-	82.1	60.2	69.4	84.2	70.2	76.5
Bao et al. (2020) (BERT 12 layers)	77.6	**63.3**	69.7	-	-	-	-	-	-	46.5	37.9	41.7	-	-	-	-	-	-
Cheng et al. (2020) (BERT 12 layers)	-	-	-	83.1	**69.5**	**75.7**	85.9	**80.6**	**83.1**	-	-	-	82.8	**67.8**	**74.5**	85.4	**77.6**	**81.3**
BERT (4 layers)	67.8	35.2	46.4	82.6	59.0	68.8	85.2	68.9	76.2	43.2	22.4	29.5	82.4	58.0	68.1	84.8	66.9	74.8
SpellBERT (4 layers)	**83.5**	60.4	**70.1**	**83.1**	62.0	71.0	87.5	73.6	80.0	**66.0**	**47.7**	**55.4**	**82.9**	61.2	70.4	87.1	71.5	78.5
w/o graph	81.2	61.4	69.9	81.8	62.0	70.5	**87.8**	73.1	79.8	61.1	46.2	52.6	81.5	60.5	69.4	**87.5**	71.1	78.4
w/o pre-training	67.6	36.1	47.1	81.3	60.5	69.3	86.4	70.7	77.8	51.7	27.6	36.0	81.0	59.3	68.5	86.0	68.0	75.9

Table 1: Results in detection-level and correction-level. For baselines, we use their reported results. Hong et al. (2019) and Bao et al. (2020) used the Traditional Chinese corpus on csc_{14} and csc_{15}, and which makes their results incomparable to ours.

Dataset	Noises	Errors covered by confusion sets
ocr test	0/1000 (0%)	302/1303 (23.2%)
csc_{14} test	16/1062 (1.5%)	663/792 (83.7%)
csc_{15} test	10/1100 (0.9%)	605/715 (84.6%)

Table 2: Stats of datasets. Noises denote the noisy data when converting data to Simplified Chinese. The rightmost column refers to the number of errors covered by confusion set (Wu et al., 2013) on test data

Dataset	Ave Length	Time per Sent		Speedup
		Ours	12-layer BERT	
ocr test	10.2	48	76	1.58x
csc_{14} test	50.0	98	153	1.56x
csc_{15} test	30.6	77	119	1.54x

Table 3: Speed comparison (ms/sentence). Preprocessing time is excluded. We set batch size as 1 and do experiments on 4 cores of a Intel(R) Xeon(R) Silver 4114T CPU following Hong et al. (2019).

as many as ours. However, by fusing pinyin and radical features and the feature-rich pre-training, SpellBERT still has the best performance on the OCR dataset. Compared with Nguyen et al. (2020), our work has better results on both csc_{14} and csc_{15}. However, there is still a gap between SpellBERT and SpellGCN (Cheng et al., 2020) which means that our devised extra modules can not completely make up for the reduced size of model.

Effectiveness of Modules We also remove graph and pre-training stage respectively to test their effectiveness. The results showed that pre-training can generally bring significant improvements on all datasets which suggests that pre-training is an effective way on CSC. The contribution of the graph mechanism is not that impressive but this makes it possible to only transfer our encoder parameters to other architectures.

Impact of Confusion Set Notice that our improvements over previous work are more obvious on ocr than that on csc_{14} and csc_{15}. Firstly, there are inevitable noises when converting data into Simplified Chinese and the noisy ratio is 1.5% and 0.9% for csc_{14} and csc_{15}. The other reason is that previous work such as Nguyen et al. (2020) and Bao et al. (2020) relied on confusion set to filter candidates. 83.7% and 84.6% of test errors in csc_{14} and csc_{15} are covered by confusion set which is an ideal and infrequent situation. On ocr which has much fewer errors covered by confusion set, they naturally performed worse.

On ocr, confusion set can simply cover 23.2% of errors and the average length of sentences are much shorter. The confusion set can be viewed as out-of-domain on ocr. SpellBERT substantially improves the performance on this dataset which indicates that SpellBERT can generalize well on different corpus without dependence on confusion set. The ablation studies further demonstrate that our proposed modules help deal with unseen errors.

Efficiency Analysis With only half the number of parameters of a 12-layer BERT, SpellBERT has the best space efficiency compared to BERT-based work. To verify time efficiency, we incorporate a speed measure in terms of absolute time consumption per sentence mentioned in Hong et al. (2019). Results in Table 3 indicate that SpellBERT can speed up at least 1.5 times.

5 Conclusion

In this work, we propose a lightweight pretrained model, SpellBERT, for Chinese spelling check. We incorporate pinyin and radicals as phonetic and visual features and design two pre-training tasks to encourage the pre-trained model to explicitly capture erroneous patterns. Experiments show that SpellBERT has competitive performance compared to the large pretrained models. Besides, SpellBERT can be directly used without confusion set in the fine-tuning and inference phase, which is more

convenient to use and easier to deal with the errors uncovered by the existing confusion sets.

Acknowledgement

This work was supported by the National Key Research and Development Program of China (No. 2020AAA0108702) and National Natural Science Foundation of China (No. 62022027).

References

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

Zuyi Bao, Chen Li, and Rui Wang. 2020. Chunk-based chinese spelling check with global optimization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2031–2040. Association for Computational Linguistics.

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

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. 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, ACL 2020, Online, July 5-10, 2020*, pages 871–881. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. 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@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 160–169. Association for Computational Linguistics.

Yu-Ming Hsieh, Ming-Hong Bai, and Keh-Jiann Chen. 2013. Introduction to CKIP chinese spelling check system for SIGHAN bakeoff 2013 evaluation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 59–63. Asian Federation of Natural Language Processing.

Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao. 2021. PHMOSpell: Phonological and morphological knowledge guided Chinese spelling check. 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)*, pages 5958–5967, Online. Association for Computational Linguistics.

Chong Li, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2021. Exploration and exploitation: Two ways to improve Chinese spelling correction models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 441–446, Online. Association for Computational Linguistics.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and phonologically similar characters in incorrect simplified chinese words. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 739–747. Chinese Information Processing Society of China.

Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. 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)*, pages 2991–3000, Online. Association for Computational Linguistics.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Charbert: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 39–50. International Committee on Computational Linguistics.

Minh Nguyen, Gia H. Ngo, and Nancy F. Chen. 2020. Adaptable filtering using hierarchical embeddings for chinese spell check. *CoRR*, abs/2008.12281.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing*

Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 8024–8035.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 43–54. Association for Computational Linguistics.

Xipeng Qiu, TianXiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *SCIENCE CHINA Technological Sciences*, 63(10):1872–1897.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3660–3670. International Committee on Computational Linguistics.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to SIGHAN 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, pages 32–37. Association for Computational Linguistics.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. 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, October 31 - November 4, 2018*, pages 2517–2527. Association for Computational Linguistics.

Dingmin Wang, Yi Tay, and Li Zhong. 2019a. Confusionset-guided pointer networks for chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785.

Dingmin Wang, Yi Tay, and Li Zhong. 2019b. Confusionset-guided pointer networks for chinese spelling check. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5780–5785. Association for Computational Linguistics.

Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J. Smola, and Zheng Zhang. 2019c. Deep graph library: Towards efficient and scalable deep learning on graphs. *CoRR*, abs/1909.01315.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at SIGHAN bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 35–42. Asian Federation of Natural Language Processing.

Bright Xu. 2019. Nlp chinese corpus: Large scale chinese corpus for nlp.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, listen, and see: Leveraging multimodal information helps Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 716–728, Online. Association for Computational Linguistics.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, October 20-21, 2014*, pages 220–223. Association for Computational Linguistics.

Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. 2014. Overview of SIGHAN 2014 bake-off for chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, October 20-21, 2014*, pages 126–132. Association for Computational Linguistics.

Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan Wang, Zhongjun He, Yu Sun, Hua Wu, and Haifeng Wang. 2021. Correcting Chinese spelling errors with phonetic pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2250–2261, Online. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 882–890. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: enhanced language representation with informative entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1441–1451. Association for Computational Linguistics.