Max Local Entropy Error Generation for Semantic Spelling Correction in Chinese

Anonymous ACL submission

Abstract

Chinese spelling correction (CSC) is a task to detect and correct spelling errors in Chinese texts. Some Chinese spelling errors are semantic errors, which can not be corrected only depending on syntax rules and local context. Global semantic information is needed to correct these errors. BERT-based models have proven to be an effective way to do CSC task. However, due to a lack of semantic errors in existing datasets, the BERT's ability to capture global semantic information is weakened. This causes the models' vulnerability to real-world examples. To address this, we propose a method referred to as **MLEEG** (**M**ax **L**ocal **E**ntropy **E**rror **G**eneration) to generate adversarial examples containing semantic errors. Experiment results show that BERT-based CSC models are vulnerable to adversarial examples generated by MLEEG, and adding MLEEG adversarial examples can improve the robustness of BERT-based CSC models without decreasing their performances on existing datasets.

1 Introduction

Chinese Spelling Correction (CSC) aims to detect and correct spelling mistakes in Chinese texts (Yu and Li, 2014; Xiong et al., 2015; Wang et al., 2019). It has received steady attention over the past two decades (Chang, 1995; Xin et al., 2014; Hong et al., 2019; Li et al., 2021). The errors in Chinese texts can be generally divided into two types: syntax errors and semantic errors. Syntax errors do not make the semantics ambiguous and thus can be corrected depending on the several characters around the error position. However, the correction of semantic errors needs the global semantic information of the sentence. Consequently, syntax errors usually show a strong local dependence, while semantic errors exhibit global dependence.

Table 1 illustrates an example of syntax error and semantic error separately. The first case is a semantic error. The semantic clue for the correction from "轲" (a Chinese name) to "课" (class) is the word "同学" (classmates) which is located seven characters after the error. We can regard this clue as global semantic information. The second case is a syntax error, for the mistake of "我门" does not make the semantics ambiguous, and we can deduce the correction depending on the local context: "但我门相信".

wrong correction
W: 下轲(ke,1)以后我建议跟同学一起去唱歌。 I suggest going to sing with classmates after Khorne.
R: 下课(ke,4)以后我建议跟同学一起去唱歌。 I suggest going to sing with classmates after class.
P: 下再(zai,4)以后我建议跟同学一起去唱歌。 I suggest going to sing with classmates down after next.
right correction
W: 但我门相信您会处理得很好. But door and I believe that you can tackle it.
R=P: 但我们相信您会处理得很好. But we believe that you can tackle it.

Table 1: Instances from SIGHAN (Wu et al., 2013). The first case is a wrong correction and the second case is a right correction. **W** represents the sentence with errors, **R** represents the gold target and **P** represents the correction made by BERT-finetune correction model.

Currently, the state-of-the-art models in CSC tasks are based on BERT (Devlin et al., 2018; Zhang et al., 2020; Cheng et al., 2020; Kipf and Welling, 2016; Zhang et al., 2020; Cheng et al., 2020), which is pre-trained with masked language modeling. However, we find that the actual datasets do not contain sufficient semantic errors. This makes the correction networks overly dependent on the local context which contains only several worlds around the error. Models trained or evaluated with only these data may not perform well on real-world tasks where semantic errors are frequent, and global semantic information is important. Table 1 contains a right correction and a wrong correction made by the BERT-finetune CSC model. We find that the BERT-finetune CSC model succeeds in the second case but fails in the first

one. This indicates that BERT-based CSC models trained on the existing datasets are more vulnerable to semantic errors.

To address this, we propose a workflow to generate semantic errors automatically. We create an adversarial text generation method called **MLEEG** (**M**ax **L**ocal **E**ntropy **E**rror **G**eneration). We first define a local entropy for Masked Language Modeling in this method. Then we generate training and evaluating examples by substituting the character of the largest local entropy. Spelling errors generated in this way are difficult to correct based on the local context, for the relatively large local entropy makes BERT-based correction networks confused and unable to choose the right character from the candidates. Consequently, global semantic information is required to make the correction, and these errors are basically semantic errors. By including **MLEEG** training data, we hope to alleviate the models' over-dependence on the local context and strengthen their ability to apply semantic information to spelling correction.

Through experimentation, we show that our examples generated by MLEEG method can improve the robustness of BERT-based CSC models without causing performance degradation on existing data. This demonstrates the effectiveness of our MLEEG method.

2 Method

Here we describe a method to automatically generate CSC examples containing semantic errors from a real-world sentence that is considered as correct. Our MLEEG method has mainly two steps. In the first step, we seek the position of max local entropy in a given sentence. In the second step, once the position is determined, we substitute the character with a random character from the confusion set.

2.1 Problem Formulation

Chinese Spelling Correction (CSC) aims to correct all the misspelled characters in the input sentence. Given an input sentence $X = \{x_1, x_2, \cdots, x_n\}$ of n characters which may contain spelling mistakes, the model needs to generate an equal-length sentence $Y = \{y_1, y_2, \cdots, y_n\}$ where all the mistakes are expected to be detected and corrected.

2.2 Max Local Entropy Strategy

For masked language modeling, the entropy of the k-th character in a sentence is defined as Equation 1:

$$\mathrm{H}(X_k) = \sum_i -\mathrm{P}(x_k^i)\ln(\mathrm{P}(x_k^i)) \quad (1)$$

where $\mathrm{H}(X_k)$ denotes the entropy of the k-th position in the sentence, and $\mathrm{P}(x_k^i)$ denotes logit output of character i in the k-th position.

Here we calculate local entropy for masked language modeling by windowing the neighboring areas of the masked position and taking this windowed part as the input of BERT masked language modeling. We can formalize the local entropy of the k-th character in a sentence by Equation 2:

$$\mathrm{H}_{\mathrm{local}}(X_k, w) = \sum_i -\mathrm{P}(x_k^i|X_w^k)\ln(\mathrm{P}(x_k^i|X_w^k)) \quad (2)$$

where $\mathrm{H}_{\mathrm{local}}(X_k, w)$ denotes the local entropy of the k-th position the sentence using window size w, and $\mathrm{P}(x_k^i|X_w^k)$ denotes the logit outpucorrectcter i in the k-th position when we take the windowed part around position k of window size w as the input of BERT masked language modeling. For BERT-based CSC models, the larger the local entropy is, the more difficult it is to correct the error depending on the local context.

Given a real-world sentence, we first tag the positions containing the person's name, time, quantitative words, and auxiliary words using POS (Part-Of-Speech) tagging. These positions usually have large local entropy but there are usually many acceptable corrections. Consequently, we will not choose these positions during the generation. Then we mask each time a position in the sentence except the tagged positions and calculate the local entropy of this position. Finally, we choose the position with the largest local entropy.

2.3 Substitution Strategy

We know that almost all Chinese spelling errors are caused by the misuse of similar valid characters in the vocabulary (Kukich, 1992; Jia et al., 2013; Yu and Li, 2014). Consequently, we apply a confusion set containing phonologically similar characters or visually similar characters to make the substitution (Liu et al., 2021).

Once the replacement position is determined, we replace it with a character randomly selected from its confusion set (90% of the time) or a random Chinese character (10% of the time).

3 SGHC Dataset

The texts in SIGHAN are collected from the essay section of Test of Chinese as a Foreign Language. These data lack informal daily Chinese texts which are frequent in the application of CSC tasks. To improve the diversity of datasets, we generate a dataset referred to as SGHC for CSC task based on the comments sampled from Sophera (a Chinese beauty forum), Gamersky (a Chinese game forum), and Hupu (a Chinese sports forum). Then we generate adversarial examples using the MLEEG method we mentioned in Section 2.

Randomly generated Data We first sample comments from three Chinese forums and split them into sentences according to punctuation that indicates the ending of a sentence. Then we remove the sentences with non-Chinese characters and also the sentences of improper length (under 7 characters or above 64 characters). After that, we tag the positions containing the person's name, time, quantitative words, and auxiliary words using POS (Part-Of-Speech) tagging. Then we randomly select a substitution position in each sentence avoiding the tagged positions and replace the selected character using the same substitution strategy mentioned in Section 2. Finally, we separate the examples into a training set and evaluating set.

MLEEG Data Based on the SGHC dataset, we select 60000 training examples whose length is above 20 characters. We then generate 60000 adversarial examples by MLEEG method with a window size w of 5, and combine them with the randomly generated data to get $\mathrm{SGHC}_{\mathrm{MLEEG}}$ training data. We then select all the evaluating data whose length is above 20 characters to generate $\mathrm{SGHC}_{\mathrm{MLEEG}}$ evaluating data by MLEEG method with a window size w of 5. The statistics of the dataset are shown in Table 5 in Appendix A.

4 Experiment and Results

In this section, we seek to understand how well BERT-based CSC models trained on normal datasets perform on MLEEG errors. We then verify that adding the MLEEG adversarial examples can improve the robustness of BERT-based CSC models.

4.1 Dataset and Evaluation Metrics

Training and evaluating Data In the experiment on SIGHAN, our training data consists of human-annotated training examples from SIGHAN 13 (Wu et al., 2013), SIGHAN14 (Yu et al., 2014), SIGHAN15 (Tseng et al., 2015), and 271K training examples generated automatically by ASR- and OCR-based method from Wang et al. (2018). We use the test set of SIGHAN13, SIGHAN14, and SIGHAN15 for evaluation. We also conduct an experiment on SGHC which is proposed ourselves. The statistics of the datasets used are shown in Table 5 in Appendix A.

Evaluation Metrics We evaluate the models' performance by the widely adopted sentence-level precision, recall, and F1 score. In the sentence level metric, a sentence is considered to be correct if and only if all errors in the sentence are corrected to the corresponding word in the target sentence.

4.2 Models and parameter settings

In this paper, we conduct experiments on the BERT-finetune CSC model. In the experiment on SIGHAN, We first finetune the model on Wang et al. (2018) with a batch size of 32, and a learning rate of 5e-5. We then finetune the model on SIGHAN13, SIGHAN14, and SIGHAN15 with a batch size of 64 and a learning rate of 5e-5. In the experiment on SGHC, we set the batch size to 128, and the learning rate to 5e-5.

4.3 Main results

Table 2 shows that the BERT-finetune CSC model is vulnerable to MLEEG adversarial examples. The average drop in correction F1 score is 31.9% for SIGHAN and 15.14% for SGHC. We find that the drop on SIGHAN is more obvious. This may be caused by SIGHAN lacking semantic errors in the training set.

As expected, after adding the MLEEG adversarial examples to the training data, performance on MLEEG adversarial examples benefits a lot. Compared to the model trained on the normal training set, the average improvement of correction F1 score is 2.3% for SIGHAN and 6.62% for SGHC.

Meanwhile, the performances on the normal test data do not suffer a decrease. The average change of correction F1 score is +0.15% for SIGHAN and +0.34% for SGHC. This proves that our MLEEG adversarial data can improve the robustness of BERT-based CSC models on semantic errors without decreasing their performance on existing data.

	SIGHAN13 correction			SIGHAN14 correction			SIGHAN15 correction			SGHC correction		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
$\mathbf{D} \rightarrow \mathbf{T}$	83.11	71.55	76.90	**57.56**	**61.84**	**59.62**	68.68	**72.69**	70.63	68.20	**61.62**	64.74
$\mathbf{D}_{\mathbf{MLEEG}} \rightarrow \mathbf{T}$	**83.39**	**73.10**	**77.91**	56.16	60.67	58.33	**70.95**	71.75	**71.35**	**69.24**	60.86	**64.78**
$\mathbf{D} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$	38.51	23.91	29.51	42.06	36.34	38.99	46.49	39.60	42.77	52.99	46.62	49.60
$\mathbf{D}_{\mathbf{MLEEG}} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$	**42.66**	**27.66**	**33.56**	**43.64**	**37.98**	**40.61**	**47.93**	**40.74**	**44.00**	**60.34**	**52.63**	**56.22**

Table 2: Performance of BERT finetune CSC model on normal examples and MLEEG examples. In the left column of the table, the left side of the arrow indicates the training data and the right side indicates the evaluating data. For example, $\mathbf{D}_{\mathbf{MLEEG}} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$ indicates that the model is trained on data enforced by MLEEG examples and evaluated on MLEEG examples, and $\mathbf{D} \rightarrow \mathbf{T}$ indicates that the model is trained and evaluated on the normal dataset (SIGHAN or SGHC). In order to offset the effect of training data size, randomly generated examples are added to $\mathbf{D}$, keeping $\mathbf{D}$ and $\mathbf{D}_{\mathbf{MLEEG}}$ have the same number of training examples. The P, R, and F1 denote separately the precision, recall, and F1 score in the sentence level and in the correction stage.

	SIGHAN15 correction			SGHC correction		
	P	R	F1	P	R	F1
$\mathbf{D} \rightarrow \mathbf{T}$	68.68	**72.69**	70.63	68.20	**61.62**	64.74
$\mathbf{D}_{\mathbf{MLEEG}} \rightarrow \mathbf{T}$	**70.95**	71.75	**71.35**	**69.24**	60.86	**64.78**
$\mathbf{D}_{\mathbf{w/oPOS}} \rightarrow \mathbf{T}$	68.59	69.49	69.04	66.32	59.66	62.81
$\mathbf{D} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$	46.49	39.60	42.77	52.99	46.62	49.60
$\mathbf{D}_{\mathbf{MLEEG}} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$	**47.93**	**40.74**	**44.00**	**60.34**	**52.63**	**56.22**
$\mathbf{D}_{\mathbf{w/oPOS}} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$	41.31	34.24	37.44	56.41	49.62	52.80

Table 3: The results of ablation experiment. $\mathbf{D}_{\mathbf{w/oPOS}} \rightarrow \mathbf{T}_{\mathbf{MLEEG}}$ indicates that the model is trained on data enforced by MLEEG examples without using POS(Part-Of-Speech) tagging and evaluated on MLEEG examples.

4.4 Ablation experiments

In this section, we conduct an ablation study on SIGHAN15 and SGHC by removing the POS (Part-Of-Speech) tagging part in our MLEEG method. The results are represented in Table 3. We can see that when we remove the POS tagging, the performances (correction F1 scores) on both datasets drop. On SIGHAN15, the performance of the model trained on examples generated by MLEEG without POS tagging is even worse than the model trained on the normal dataset. This fully demonstrates the effectiveness and necessity of using POS tagging.

5 Case Study

In this section, we give three cases we generate from MLEEG in Table 4 . In these cases, there are many possible corrections according to the syntax. Global semantic information is thus needed to determine the correction. In the first case, the correction from "答" to "他" which is a pronoun needs the name information at the beginning of the sentence. In the second case, the correction from “瞿” to “去” concerns the determination of a verb, which need the semantic information of the third character: "去". In the third case, the correction from “力” to “大” concerns the determination of a noun, which need the semantic information of "大学生". These cases show that the MLEEG method can generate semantic errors in various lexical categories.

case 1
RIGHT:张爱文很聪明，老师教他英文、地理什么的，他很快明白了。 Zhang is really smart, the teachers teach him English, Geography etc. He understands quickly.
MLEEG:张爱文很聪明，老师教答英文、地理什么的，他很快明白了 Zhang is really smart, the teachers teach and answer him English, Geography etc. He understands quickly.
case 2
RIGHT:我想去你的家了，可是我不能去，因为今天我有大的考试。 I want to go to your home, but I can not go, because I have a big exam.
MLEEG:我想去你的家了，可是我不能瞿，因为今天我有大的考试。 I want to go to your home, but I can not Qu, because I have a big exam.
case 3
RIGHT:我是支持大学生半工半读的人。为什么? 因为大学生出去打工，每个月可以多赚一笔钱。 I am a supporter of half-time university students. Why? Because university students can earn an extra sum of money every month by going out to work.
MLEEG:我是支持大学生半工半读的人。为什么? 因为力学生出去打工，每个月可以多赚一笔钱。 I am a supporter of half-time university students. Why? Because power students can earn an extra sum of money every month by going out to work.

Table 4: Cases of the errors we generate by the MLEEG method. The sentence after **RIGHT** is a gold target sentence in SIGHAN datasets, and the sentence after **MLEEG** is an example generated by MLEEG method. Errors in the adversarial examples are in red. The corresponding corrections in the gold target are in blue. Semantic information to correct the errors is in magenta.

6 Conclusion

In fact, Chinese spelling errors can be divided into two categories: syntax errors and semantic errors. We proposed accordingly an adversarial example generation method, referred to as MLEEG to generate semantic errors. Experiment results demonstrate that BERT-based CSC models are vulnerable to MLEEG adversarial examples, and including MLEEG adversarial examples can improve the robustness of BERT-based CSC models to real-world examples.

7 Limitations

We found that some semantic errors can have different ways to correct them. This is usually because the context is not specific enough to uniquely determine the correction. We have applied POS (Part-Of-Speech) tagging to decrease these examples during the generation, but this may not be enough. How to further decrease the generation of these errors is worth further study.

References

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. *arXiv preprint arXiv:2004.14166*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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 2019)*, pages 160–169.

Zhongye Jia, Peilu Wang, and Hai Zhao. 2013. Graph model for chinese spell checking. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 88–92. Asian Federation of Natural Language Processing.

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439.

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, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 441–446. Association for Computational Linguistics.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2991–3000. Association for 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*, pages 32–37, Beijing, China. 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*, pages 2517–2527, Brussels, Belgium. Association for Computational Linguistics.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. 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.

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

Yang Xin, Hai Zhao, Yuzhu Wang, and Zhongye Jia. 2014. An improved graph model for chinese spell checking. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, October 20-21, 2014*, pages 157–166. Association for Computational Linguistics.

Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng. 2015. Hanspeller: A unified framework for chinese spelling correction. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 20(1).

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, pages 126–132, Wuhan, China. 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.

A Figure

Training Data	sent	Avg.Length	errors
SGHC	109985	25.27	109985
$\text{SGHC}_{\text{MLEEG}}$	169486	28.50	158842
SIGHAN13	700	41.81	343
SIGHAN14	3437	49.55	5136
SIGHAN15	2339	31.32	3048
Wang et al. (2018)	271329	42.54	381962
Test Data	**sent**	**A vg.Length**	**errors**
SGHC	600	24.87	600
$\text{SGHC}_{\text{MLEEG}}$	318	34.16	291
SIGHAN13	1000	74.32	1224
SIGHAN14	1062	50.01	771
SIGHAN15	1100	30.64	703

Table 5: Statistics of dataset