NLP Final Report Team 5

Xuan Zhao
xz3633@nyu.edu

Victor Cui
vyc8567@nyu.edu

Yuhe Tan
yt2336@nyu.edu

Abstract

This project replicates the results from the paper "A Sentence Classification Method for Chinese Spelling Error Detection Based on BERT". (Jiang and Zhou, 2021) The paper conducts spelling error detection (SED) at sentence-level and purposes two training data shuffling methods called "Single-shuffle" and "Pair-shuffle" in order to improve the model's performance. We fine-tune BERT (Devlin et al., 2018) and ELECTRA (Clark et al. [2020]) to do SED with Chinese SIGHAN dataset using the two proposed shuffling methods in addition to the standard method. Besides, a new English dataset is used to finetune on two models to test whether the models are more suitable for English dataset on SED. Finally, we try to do spelling error correction (SEC) directly using BERT-finetune to compare with SED performance.

1 Introduction

Spelling Error Check is an essential task in natural language processing. In particular, Chinese Spell Checking (CSC) aims to detect and correct Chinese spelling errors. Chinese spelling errors frequently arise from confusion among multiple-character words that are phonologically and visually similar but semantically distinct.

Recently, deep learning-based models have gradually become the mainstream CSC methods, especially with the emergence of powerful pre-trained models such as BERT (Devlin et al., 2018) and ELECTRA (Clark et al., 2020). We were interested in Chinese Spell Checking using pre-trained models, and the paper "A Sentence Classification Method for Chinese Spelling Error Detection Based on BERT" (Jiang and Zhou, 2021) was found to meet our interest. Most Chinese spelling error tasks aim at detecting and correcting errors at word-level and character-level. However, the paper proposes the character-level detection is too strict for a checker, and a sentence-level checker also has significant meaning in exploring whether the model can learn the sentence-level information inside the labeled data. Besides, two shuffling methods "Single-shuffle" and "Pair-shuffle" are proposed to improve the SED performance. After reading the paper, we are curious about the effectiveness of the two shuffling methods in sentence-level SED tasks. If we could find them effective through our own experimentation, we could try to use them on other related NLP tasks.

In this project, we replicates the main results of this paper: fine-tuning BERT on the Chinese SIGHAN dataset using three different shuffling methods, "original-shuffle", "single-shuffle", and "pair shuffle". Besides, we performs three extensions related to SED. Extension 1 focuses on using ELECTRA model doing the same task as in the paper. Extension 2 uses an English dataset to finetune on BERT and ELECTRA model. Extension 3 does SEC on the same SIGHAN dataset as in the paper with fine-tuning BERT.

2 Experiments Setup

2.1 Dataset Construction

The original paper uses SIGHAN dataset, a benchmark for CSC. A sentence with or without spelling errors is given as the input. The labels are the locations of incorrect characters and the correct characters. Each character or punctuation mark occupies one spot for counting location. The dataset is directly accessible though the paper's GitHub repository (Jiang, 2020). We perform three different shuffling methods on the dataset:

Original-shuffle: The original shuffle method is the standard method used in BERT pre-training and fine-tuning. Each correct sentence appears only once, along with multiple wrong counterparts of it each with different misplaced tokens. Dataset batch shuffling is completely random.

Single-shuffle: For each misplaced tokens in the same sentence, a correct sample and its wrong counterpart is appended to the training data, resulting in correct sentences appearing multiple times in the dataset. Dataset batch shuffling is again completely random.

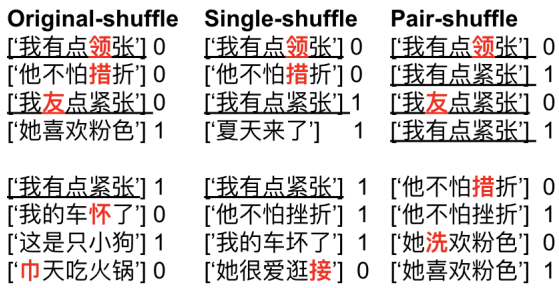

Figure 1: Example of three shuffle methods

Pair-shuffle: Similar as "single-shuffle", but when shuffling, the correct and wrong sentences for a training sample are bind together in one batch when creating batches.

Figure 1 shows an example of the model inputs with three shuffling methods. Color red indicates the wrong word in each sentence. "0" and "1" are the labels for the input. Same as paper, "0" represents the wrong sentence, "'1" represents the correct sentence.

2.2 Replication setup

In the paper, the pair-shuffle and single-shuffle methods make noticeable difference on performance of BERT fine tuning, and the fine-tune BERT with original dataset is the main comparison to show whether the shuffle methods work. Thus, we plan to fine-tune BERT using SIGHAN dataset three times, with original-shuffle (Baseline), single-shuffle, and pair-shuffle. The chosen BERT model is "bert-base-chinese" with Adam optimizer (Cui et al., 2021), lr=2e-6, betas=(0.9, 0.999), eps=1e-8. These hyper-parameters are same as paper. Then we perform evaluation of these three fine-tuned models on a held out test set from SIGHAN.

2.3 Extension 1 setup

Unlike BERT that is mainly trained as MLM, ELECTRA(Clark et al., 2020), a model designed by Google, is instead trained by corrupting original sentence with incorrect tokens, which we think aligns more with the nature of SED task. Therefore we plan to replace BERT with ELECTRA in our replication of the paper as the first extension. We decide to evaluate two model sizes, "electra-small" and "electra-base", where "electra-base" has the same size as the "bert-base" model in replication. The small model is chosen because it is easier to train and optimize than the base. Note original ELECTRA is pretrained with a generator and a discriminator combined, but in our downstream SED task we only make use of the discriminator part, which was the part pre-trained with detecting corrupted/replaced tokens, with a fully connected part on top for sequence classification.

In our experiment, We used models named "hfl/chinese-electra-180g-small-discriminator" and "hfl/chinese-electra-180g-base-discriminator" from Huggingface. (Cui et al., 2021) Both models are pre-trained using Chinese data from scratch, and has same sizes with their English counterparts from Google. (Clark et al., 2020) Because, unlike for replication on BERT, we do not have information on the hyper-parameters of the model, we perform Bayesian hyper-parameter search. Due to resource constraint, we manually picked epoch=20, batch size=64, and only search for learning rate, which is 6e-6 for small and 2e-6 for base. Note our search for base is still far from optimal, so it can potentially hinders results. We perform evaluation on the same held-out dataset from SIGHAN as in replication.

2.4 Extension 2 setup

Since the original dataset used is in Chinese, we want to evaluate the effectiveness of the proposed shuffle techniques on an English dataset. The English dataset is gotten from spellcorrect (Bhoosreddy, 2013). It has a large corpus file called "corpus.data" and a words file called "spellerrors.data" which contains right words and corresponding wrong words, e.g. "raining: raning". We split this corpus into sentences, and replace the right word in the sentence using wrong word in "spellerrors.data". Since this corpus is really large, we just create one wrong sentence for each right sentence. This gives us 273949 training sentences. Similar to replication and extension 1, we are going to fine-tune BERT and ELECTRA on the English dataset for SED. To save time, we re-used the same hyper-parameters. The models used are English version BERT-base ("bert-base-uncased") (Devlin et al., 2018), English version ELECTRA-small("google/electra-small-discriminator"), and English-version ELECTRA-base ("google/electra-base-discriminator) (Clark et al., 2020). Again, we only use the discriminator parts in ELECTRA as we did in extension 2.

Table 1: Main results

Method	Acc.	Prec.	Rec.	F1.
Original-Shuffle				
BERT (paper)	80.0	73.7	73.2	73.5
BERT (repl.)	84.1	81.0	89.1	84.9
Electra-small	81.6	80.1	82.2	76.4
Electra-base	84.2	83.8	85.7	82.0
Single-Shuffle				
BERT (paper)	82.6±0.4	78.6±0.9	89.5±0.9	82.6±0.2
BERT (repl.)	83.5	79.6	90.0	84.5
Electra-small	81.4	79.7	87.6	73.1
Electra-base	83.8	83.2	86.5	80.2
Pair-Shuffle				
BERT (paper)	83.4±0.4	80.1±0.7	88.9±0.6	84.3±0.3
BERT (repl.)	83.3	79.8	89.1	84.2
Electra-small	82.6	79.9	86.9	73.7
Electra-base	84.5	84.0	86.5	81.6

2.5 Extension 3 setup

The SIGHAN dataset used in replication and extension 1 contains wrong words and positions as labels, so we can use this same dataset to do spelling error correction. The model is same as replication setup using BERT-finetune with "bert-base-chinese" but with lr=2e-7 and the default Adam parameters.

3 Results and Discussion

3.1 Replication

Taking paper's code as reference (Jiang, 2020), we found one mistake in their codes: they used "re.sub" function. For example, the wrong sentence is "He took the bag and went into the bag.", and suppose that the last word is incorrect. It should be "bar" instead of "bag". If we use "re.sub", the sentence will become "He took the bar and went into the bar." which is not the right sentence. We found that if we use "re.sub", there will be 5873 sentences that are wrongly changed, which is a pretty large number since there are only 29354 training inputs for "single-shuffle" and "pair shuffle", and 14881 for "original-shuffle". After fixing the issue, we got the final result as shown in the Table 1. We don't know where the results of BERT-Finetune in the original paper came from. It did not mentioned in the paper nor in the codes, so we just used the same hyper-parameters as paper but with "original-shuffle". Unfortunately, as shown by our replication data, the "single shuffle" and "pair shuffle" did not improve the performance. However, one thing need to notice is that the SIGHAN dataset used by original paper contains lots of duplicate data. "Original-shuffle" actually removes those duplicated data, but "single-shuffle" and "pair-shuffle" implemented in the paper does not. We believe removing duplicated data to keep consistency is the correct way to do comparisons between methods, thus, in our own replication of "single-shuffle" and "pair-shuffle" experiments, we removed duplicated data. Both of them got test accuracy around 84, so these two shuffle methods cannot improve the performance nor worsen it. Furthermore, we use "re.sub" code to do "original-shuffle" again to learn the influence of it and obtain test accuracy around 83.1, which is lower than 84.1, but we still don't know how the paper got 80.0.

3.2 Extension 1

The results of the evaluation of our fine-tuned ElECTRA models are shown in Table 1. There, again, are not enough evidence to prove the effectiveness of the two proposed shuffling methods in the original paper. So our discussion for this extension will be mostly focused on the difference between BERT and ELECTRA models in general. "electra-base" has slightly better results than same-sized BERT model, though the differences are not significant. It also has to be taken into consideration that the fine-tuning of our "electra-base" model is not optimized, so the actual potential of the model may be higher than what is shown in our results. Therefore, we believe our results suggest ELECTRA has a small advantage on Chinese SED tasks than BERT.

In addition, it's clear that "electra-small" shows competitive performances on Chinese SED compared to BERT models and larger sized ELECTRA model across all experiments. Thus, considering it requires much less time, computing, and storage resources, we think it should be a very viable, light-weight alternative for SED tasks.

3.3 Extension 2

We test all of our models on an English dataset since both BERT and ELECTRA are designed with English from beginning. As we mentioned in Extension 2 setup, we create only one wrong sentence for each right sentence, so there is no difference between "original-shuffle" and "single-shuffle". Also our replication results show that two proposed shuffle methods did not work, so we just do the normal original shuffle. Results are shown in Figure 2. The plot shows that both

BERT-base and Electra-base perform well better detecting spelling error on English than on Chinese. The results of Electra-small is also higher than the result on Chinese dataset, and they are slightly lower than those of larger sized models, but justifiable for a light-weight option. However, since our English dataset is much larger than Chinese dataset, we cannot be sure that this huge difference between performances on Chinese and English SED tasks across all models is caused by the models, the data size and quality, or the nature of the two languages.

3.4 Extension 3

As "single-shuffle" and "pair-shuffle" did not work, in this last extension we shift our focus on SEC. We are curious about spelling error correction accuracy on BERT-finetune. Results are shown in Figure 3. The pink bar is the result we got in the replication part with BERT-finetune (original-shuffle). The dark blue bar is the result of spelling error correction. The plot shows that the spelling error correction can get similar even higher accuracy than sentence level error detection. Thus, we think that it's better to do spelling error correction directly.

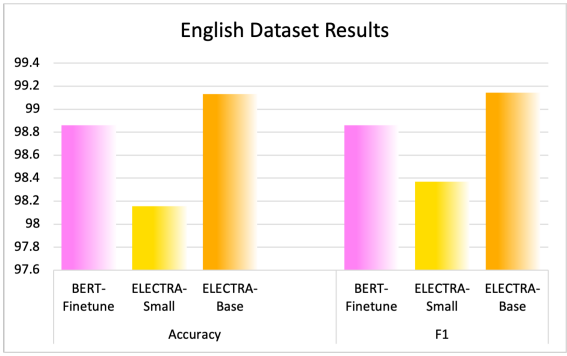

Figure 2: Extension 2 results

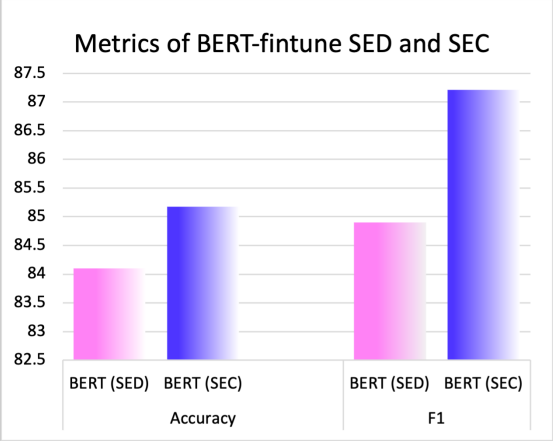

Figure 3: Extension 3 results

4 Conclusion

During our replication, the two training data shuffling techniques ("single-shuffle" and "pair-shuffle"), proposed by the original paper, are not shown to be effective in improving accuracy. In extension 1, we see that ELECTRA can be potentially more effective in SED than BERT given same model size, and smaller sized ELECTRA shows arguably good performance with less resource consumption. In extension 2, our results further hint that Chinese Spelling Error Detection on sentence level is currently a more challenging task for transformer models than English. Besides, in extension 3, we discover that Spelling Error Correction does not necessarily need a detection (on sentence level) process beforehand, as transformer models performed quite as well directly correcting misplaced tokens, making this sentence level error detection a less interesting task.

5 Limitations and Future Directions

Due to constraints, we did not fully optimize ELECTRA hyper-parameters in Extension 1, nor did on English dataset in Extension 2, so there can be potential improvements for both parts. It is also necessary to do more rounds of experiments In addition, Comparison between Chinese and English sentence level spelling error detection are not done in the same dataset, so the difficulty/quality gap between two datasets can impact our results. Specifically, the English dataset is more balanced and contains much more sentences. In the future, we can inspect more on spelling error correction, especially on Chinese since BERT and ELECTRA performs not so well on Chinese as they do on English.

6 Contribution

Xuan Zhao: Training and evaluation of Replication, Extension 2 BERT part, and Extension 3
Yuhe Tan: Dataset Construction including replicating different shuffling methods
Victor Cui: Training and evaluation of Extension 1 and Extension 2 ELECTRA part.

References

Jin Jiang and Yanquan Zhou. A sentence classification method for chinese spelling error detection based on bert. In *2021 International Conference on Asian Language Processing (IALP)*, pages 369–372, 2021. doi: 10.1109/IALP54817.2021.9675281.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. URL https://openreview.net/pdf?id=r1xMH1BtvB.

J Jiang. jiangjin1999/sentence-level-detection-on-csc, 2020. URL https://github.com/jiangjin1999/Sentence-level-detection-on-CSC.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. Pre-training with whole word masking for chinese bert, 2021. URL https://ieeexplore.ieee.org/document/9599397.

J Bhoosreddy. jbhoosreddy/spellcorrect, 2013. URL https://github.com/jbhoosreddy/spellcorrect.