Discovering the Effectiveness of Pre-trained Masked Language Models for English Spell Correction

Yunjie He[‡], **Rui Li**[†], **Wangyuan Ding**[†], **Han Zhou**[†]
[†] MSc Machine Learning
[‡] MSc Computational Statistics and Machine Learning
University College London
{yunjie.he.17, rui.li.20, ucabwdi, han.zhou.20}@ucl.ac.uk

Abstract

Misspelled words not only limit the efficiency of online communication but also degrade the performance of general natural language processing (NLP) systems. Traditional statistical spell correctors have limited ability because they are incapable of capturing contextual information. Due to BERT's context-aware property and its competitive performance on many NLP tasks, in this project, we first examine BERT's suitability in spell correction, then conduct different empirical experiments to improve BERT's performance. Based on experiment results, we propose a new model CLMBER (Char-**C**NN-**L**STM-**M**ultilingual-**BER**T model) which achieves the best performance in the spell correction experiments.

1 Introduction

Daily text entry is easily corrupted by noisy input. It inherently limits the maximal entry rate and becomes more phenomenal with smaller touchscreens for mobile devices. The out-of-vocabulary words resulted from misspelling also degrade the performance of general natural language processing systems with any word-level embedding (Mikolov et al., 2013). Therefore, it has aroused a growing body of work focusing on the researches of autocorrector for text inputs (Jayanthi et al., 2020). To provide a fast and reliable typing experience, it is crucial that the auto-corrector utilises the contextual information efficiently (Zhang et al., 2020). However, limited attention has been devoted to context-aware auto-corrector, which forms the foremost goal of this project.

Traditional corrector based on a statistical mapping from the noisy input to the dictionary fails to capture the contextual information (Formiga and Fonollosa, 2012). This lack of context inevitably leads to ambiguity once the correction is completed. In the light of the rapid development of neural networks, architectures such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) resolve the problem of dependencies and start bringing contextual patterns into account (Li et al., 2018). Recently, state-of-the-art researches have illustrated the efficiency and effectiveness of the Transformer (Vaswani et al., 2017) architecture in capturing longer-range linguistic structure. Based on that, BERT (Devlin et al., 2019) pretrains bidirectional Transformers and has shown great successes in downstream NLP tasks through fine-tuning (Conneau and Lample, 2019). With its better level of language understanding, we aim to benefit from BERT by utilising all contextual information to improve the performance of English spell correction.

Besides the importance of being context-aware, the bottleneck of neural methods relies heavily on the available training dataset. In the problem of spell correction, the amount of representative data with labels is insufficient, and they are generally difficult to procure (Qiu et al., 2020). Though generating a synthetic dataset with data augmentation is a feasible solution, the neural network is inevitably over-fitted to the training samples to some extents and leads to a relatively poor generalisation ability in dealing with real errors (Wang and Zheng, 2020). Generative pre-training (Radford et al., 2018) of a large language model, BERT, holds strong potential in overcoming this challenging situation. Hence, it becomes our primary interest in discovering the effectiveness of pre-trained BERT in spell correction.

In this paper, we firstly study the performance of BERT model in comparison to neural baselines without BERT on the task of spell correction. Then, we notice that BERT provides excellent prediction accuracy and its correction rate still has the potential to be further improved. Motivated by Zhang et al. (2020), we aim to improve the detection capability of BERT and eventually achieve a competitive word correction rate by proposing CLMBER

(Char-**C**NN-**L**STM-**M**ultilingual-**BER**T model) as the best architecture. Throughout experiments, we sequentially study the effect of fine-tuning layers, variants of BERT, and model concatenation with BERT in spell correction. From empirical results, the CLMBER model outperforms other models on correction rates and becomes our best model. By pre-training on multilingual resources and only fine-tuning on the last 3 layers, we improve the word correction rate, the percentage of corrected misspelled tokens, from 68.67% (BERT baseline) to 73.39%, and we provide corrected sentence examples that demonstrate its excellent ability in context-aware correction at the end.

2 Related work

Despite the theoretical advantages of incorporating BERT, some challenges remain to be solved for the problem of spell correction. We survey the inspiring works and discuss in detail their merits and limitations, from the traditional correction methods to state-of-the-art researches.

2.1 N-grams language model

To infer the correct text from a noisy input, traditional approaches divide this process into three sequential sub-tasks: error detection, candidate generation, and candidate ranking (Kukich, 1992). N-grams language model (Chen and Goodman, 1999) and string edit distances are then implemented to predict the most probable candidate (Islam and Inkpen, 2009). Within the n-grams model, the word is predicted according to a conditional distribution of the other $n-1$ words, $p(w_1,...,w_m) = \prod_{i=1}^{M} p(w_i|w_{i-n+1},...,w_{i-1})$, where each word is represented by w_i. More recently, Vertanen et al. (Vertanen et al., 2015) combine a probabilistic keyboard model with a 4-grams language model on sentence-based decoding. It is evident that more contexts can be involved when the size of the n-grams increases. However, the corpus size and training time grow exponentially with it, thereby limiting the abilities in resolving ambiguities. Hence, it motivates us to explore more context-aware approaches.

2.2 Neural spell correction

Inspired by the success of machine translation tasks, spell correction can be equivalently modeled as a translation from incorrect texts to an error-free sentence (Zhou et al., 2019). Due to the sequential property of linguistic data, RNNs become a natural choice of language model and lead to neural machine translation solutions. Either LSTM or gated recurrent unit architectures are involved in preventing the vanishing gradient problem (Bengio et al., 2003), which enable the system to cope with contextual information in translation. Within the framework of encoder-decoder RNNs, Xie et al. (Xie et al., 2016) implemented attention mechanisms to deal with noisy inputs upon a sequence-to-sequence model. Followed by that, Ghosh et al. (Ghosh and Kristensson, 2017) further improved the model by combining convolutional neural networks to capture the sub-word information. Though both of them have achieved more context-aware predictions, linguistic understanding and generalising abilities are still constrained within the limited training corpus.

2.3 Unsupervised pre-training

Unsupervised pre-training has been proved to be effective to improve the optimisation and generalisation of neural networks (Ramachandran et al., 2017). Over the last two years, phenomenal successes in NLP tasks have been accomplished by unsupervised pre-training on large language models (Radford et al., 2019). The state-of-the-art model, BERT, utilises bidirectional contexts by pre-training with the masked language model objective. During the pre-training process, it masks 15% WordPiece (Wu et al., 2016) tokens, and its loss function is built on predicting the masked token from all other tokens, thereby learning extensively on bidirectional contexts. This important feature makes it directly effective in spell correction, where the noisy input should be corrected with a good understanding of contexts.

The most recent researches have successfully implemented pre-trained language models on the task of grammatical error correction (Alikaniotis et al., 2019). Competitive results have been achieved with a relatively small quantity of annotated training dataset by fine-tuning BERT (Kaneko et al., 2020). When it comes to deal with noisy inputs, the soft-masked BERT is introduced by Zhang et al. (2020) and focuses on Chinese spell correction. It consists of an error detection network and a separate correction network to remedy the detection capability. More recently, Neuspell (Jayanthi et al., 2020) is the first toolkit that incorporates BERT to process English spell correction, and its main

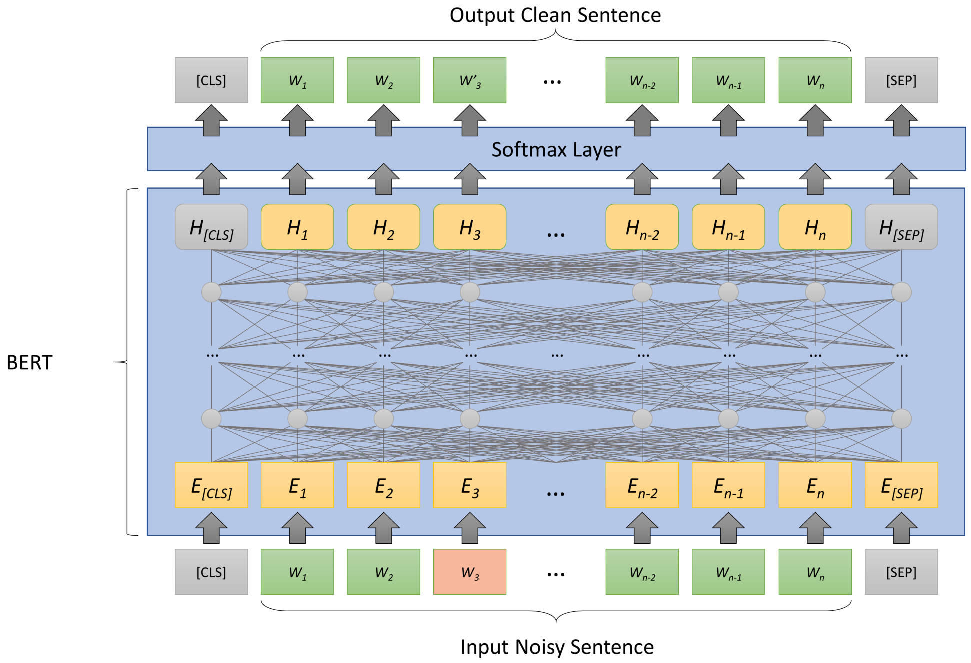

Figure 1: Subword BERT Spell Corrector. The word W_3 is corrected as W_3', and other words are copied.

deficiency is insufficient in detecting error patterns. Hence, by focusing on English spell correction tasks, our approach is extended from Neuspell and aiming to improve the detection capability. In comparison to the soft-masked BERT, our model is distinct in model architecture that contains a parallel Char-CNN-LSTM network in addition to the BERT network, which equivalently handles the role of character-level error correction and sub-word level correction. To the best of our knowledge, our work is the first study that empirically analyses each factor behind BERT and aims to improve BERT's effectiveness in English spell correction.

3 BERT for spell correction

3.1 Problem and Motivation

Spell correction can be considered as a sequence labeling task. Given the input as a sequence $X = (x_1, x_2, \ldots, x_n)$ of length n where x_i can be either characters or word depends on the tokenization algorithm, the spell correction model transforms it into another sequence of characters $Y = (y_1, y_2, \ldots, y_n)$ with the same length. Compared to other supervised downstream tasks such as Question Answering and Text Summarization, spell correction is different in the sense that only a few characters need to be corrected in a sentence, and most characters should be unchanged.

The state-of-the-art approach is to employ BERT as the Grammatical Error Corrector (Didenko and Shaptala, 2019). BERT has shown great ability to capture contextual information and acquire knowledge in Language Understanding tasks. During pretraining, one task of BERT is Masked LM, where 15% of tokens are masked and only these words are predicted. The spell correction task will benefit from the upstream Masked LM task, in the sense that wrong words can be considered as masked words and the target is to predict their corrected version. BERT has already achieved very high accuracy when testing on various datasets, thus in this paper, we will mainly improve the word correction rate.

3.2 Subword Tokenization

The original subword model is called "WordPiece" (Wu et al., 2016). The principle of Subword tokenization algorithms is that common words should not be broken down into smaller subwords, but rare words should be decomposed into meaningful subwords. For example, if we consider "subword" as a rare word and decompose it into "sub" and "word". Both "sub" and "word" would appear more frequently meanwhile the meaning of "subword" is kept by the composite meaning of these two subwords.

Subword tokenization enables BERT to have a manageable vocabulary size while being able to learn meaningful representations independent of context. Furthermore, subword tokenization allows BERT to recognise unseen words by decomposing them into known subwords, thus infer their meanings.

3.3 Subword BERT

Subword BERT consists of a stack of 12 transformer blocks to take the entire subword representations $E = (E_1, E_2, \ldots, E_n)$ as input. Here the correctly spelt words are marked as themselves, and the label for the misspelled words are their correct version. For each input word, the final hidden layer of BERT $H = (H_1, H_2, \ldots, H_n)$ generates a sequence of subword representations. These representations are then averaged to obtain the word-level representations. The output of the model is then $P(y_i = j \mid X)$ which is the conditional probability generated by a softmax function that x_i is corrected as j in the vocabulary list. See figure 1 as an example.

4 Experiments and results

4.1 Experiment Setup

Datasets Due to the scarcity of available parallel data for spell correction, we use a synthetic training dataset and real-world test set (BEA60K) provided in (Jayanthi et al., 2020). The synthetic training dataset is generated by noising 20% of the tokens in 273K sentences from the one billion sentences (Chelba et al., 2014) by the "PROB" strategy in (Jayanthi et al., 2020), and it contains 2K spelling mistakes (6.1% of all tokens). The real-world test set is a combination of four small datasets, which contains 63044 sentences and nearly 70K spelling mistakes (6.8% of all tokens).

Evaluation Metrics We use correction rate (CR) to evaluate the model's performance, which is the percentage of corrected misspelled tokens. Denote the total number of tokens as N and the number of corrected misspelled tokens as M,

$$\text{Correction} \quad \text{Rates} = \frac{M}{N}.$$

Implementation Details We train all the models by considering spell correction as a sequence labelling task. Here the correctly spelt words are marked as themselves, and the label for the misspelled words are their correct version. The model's output is the probability distribution over a finite vocabulary generated by a final softmax layer for each input word. All models are trained to convergence by the Adam (Kingma and Ba, 2015) optimizer. Apart from BERT-large whose batch size is 16, the rest models are trained with batch size of 32.

4.2 Neural spell correction vs BERT

In this subsection, we compare BERT with three non-BERT-based models to see how BERT-based models perform on spell correction task:

- Char-CNN-LSTM (Kim et al., 2016): Leverage subword information through a character-level convolutional neural network, whose output is used as an input to a recurrent neural network language model.

- Char-LSTM-LSTM (Li et al., 2018): It consists of a char-level RNN and a word-level RNN. The former collects orthographic information by reading characters and the latter predicts the correct words by combining the orthographic information with the context.

- SC-LSTM (Sakaguchi et al., 2017): Based on Cmabrigde Uinervtisy (Cambridge University) effect, SC-LSTM first represents each word as a concatenation of first, last and bag of internal characters, and then correct the sentence with bi-LSTM.

The result is given in Table 1. We can see that BERT outperforms Char-CNN-LSTM and Char-LSTM-LSTM, which demonstrates BERT's effectiveness in spell correction. Although SC-LSTM outperforms BERT, given the fact that in BERT, the spell correction is only done by a linear classifier layer and SC-LSTM is explicitly designed for spell correction, BERT's performance is still promising.

Model	**CR (%)**
Char-CNN-LSTM	68.11
Char-LSTM-LSTM	67.42
SC-LSTM	**69.68**
BERT	68.67

Table 1: Comparison Result of BERT and other non-BERT baseline.

4.3 Number of fine-tuned layers

Due to the success of the BERT's performance on different NLP tasks (Devlin et al., 2019), more and more researchers use BERT as a building block in various NLP tasks (Liu, 2019; Sun et al., 2019; Adhikari et al., 2019; Nogueira and Cho, 2019) by fine-tuning it. One common way to fine-tune BERT is freezing few layers (the parameters in these frozen layers are unchanged during training) and training the model on task-specific data. Kovaleva et al. (2019) shows that only the last few layers in BERT change the most after the fine-tuning process. This evidence encourages us to consider the possibility that reducing the number of layers fine-tuned in the training process will give the same or even better accuracy in spell correction. The benefit of freezing layers is two-fold: Firstly, it prevents the trained model from overfitting, especially when the task-specific train data is small. Secondly, it brings an obvious boost to the training speed and reduces the related memory usage. In this subsection, we change the number of fine-tuned final layers in BERT (layers are frozen from bottom to top) and observe the performance. The results are given in Table 2 and Fig. 2.

Number of Frozen Layers	**CR (%)**
0	67.87
1	68.24
2	68.80
3	69.00
4	69.31
5	69.66
6	69.26
7	70.51
8	71.08
9	**71.39**
10	71.12
11	70.77
12	64.84

Table 2: Performance of BERT with different number of frozen layers.

As shown in Fig. 2, the correction rate first increases as the number of frozen layers becomes large, then decreases when the number of frozen layers is larger than 9. Note that when the number of frozen layers becomes 12 (now the whole BERT is frozen), the performance drops dramatically. This is due to the discrepancy between the BERT's training dataset and our dataset.

When the number of frozen layers is 9, we achieve the best performance. As shown in Jawahar et al. (2019), the first few layers in BERT serve the purpose of extracting general semantic features of input text, and the last few layers are task-specific. Compared with the number of parameters in BERT, our training dataset is still small. Therefore by freezing the first 9 layers, i.e. only fine-tuning the last 3 layers, the rest task-specific layers can be better fine-tuned and give better performance.

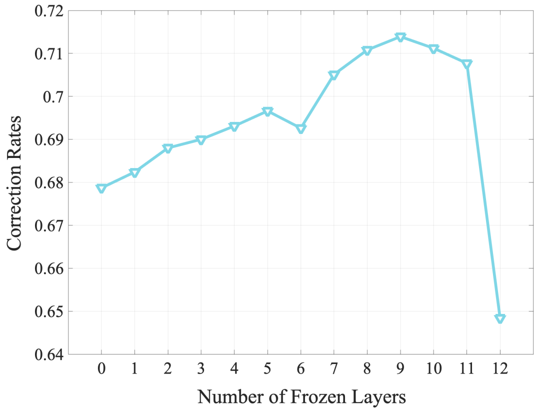

Figure 2: Performance of BERT with different number of frozen layers.

4.4 Variants of BERT

The advent of BERT marks a new era of NLP and it motivates the development of a series of its variants, including DistilBERT, multilingual BERT (mBERT), etc. In terms of their respective specialities, we analyze their performances on the spell correction task. In our experiment, we trained three variants of BERT, BERT-Large, DistilBERT, mBERT. We mainly discuss two points. Firstly, we look at the influence of model size on the effectiveness of the spell correction task by comparing BERT-Large and DistilBERT. Secondly, we find that multilingual BERT model outperforms the monolingual models by capturing the underlying association between languages.

- **BERT-Large (Devlin et al., 2019)**: is the large version of BERT-base, with 24 transformer blocks, a hidden size of 1024, 16 self-attention heads. The total number of parameters in BERT-large is around 340 millions which is three times that in BERT-base. Empirically, BERT-large is expected to increase the performance of BERT-base across all tasks, especially those with little training data.

- **DistilBERT (Sanh et al., 2020)**: is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40%

fewer parameters than BERT-base, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

- **mBERT (Devlin et al., 2019)**: Very soon after proposing BERT, Google research introduced a multilingual version of BERT capable of working with more than 100 languages.Similar to the original English BERT model, mBERT is a 12 layer transformer trained on Wikipedia pages of 104 languages with a shared piece vocabulary.

Pre-training multilingual language models, such as mBERT and XLM, have pushed the state-of-the-art on the cross-lingual NLP tasks, including question answering and named entity recognition by jointly pre-training large transformer models on many languages. The team of Facebook AI (Conneau et al., 2020) had a comprehensive study of important factors that are influential to large pre-training scale multilingual models. This study suggests the trade-off between capacity dilution (Arivazhagan et al., 2019) and positive transfer in the pre-training process. In simple words, the capacity dilution means the increase in the number of languages in the pre-training stage is likely to induce the decrease in per-language capacity given fixed model size. This phenomenon is named as *the curse of multilinguality*. On the other hand, the model is also able to take benefits from such an increase because low-resource language performance can be improved by adding similar higher-resource languages during pre-training. This advantage is called positive transfer. The team shows that the capacity of multilingual models to leverage training data coming from multiple languages for a particular task can overcome the capacity dilution problem to obtain better overall performance as long as we keep the number of languages in the pre-training stage within a reasonable range. The obtained overall performance beats that achieved by the monolingual BERT model.

This result motivates us to apply multilingual language models on the spell correction task by regarding the misspelled sentences as a new foreign language which is very rare. In our experiment, we applied mBERT on the spell correction task and the results are listed in Table 4. It is easy to observe that the mBERT model outperforms monolingual BERT models. This result indicates that the positive transfer effect coming from the pre-training on multiple languages in mBERT can overcome the curse of multilinguality to achieve better results on the downstream task of spell correction. In order to further prove our guess, we repeated the same experiment by using multilingual DistilBERT model (mDistilBERT) (Sanh et al., 2020). An improvement is still observed in the word correction rate compared with the original DistilBERT model. Such improvement is out of the same positive transfer effect from the multilingual model.

Model	Parameters	CR (%)
mBERT	177M	**72.27**
BERT-Large	340M	70.09
BERT-base	110M	68.67
mDistilBERT	134M	68.75
DistilBERT	66M	67.91

Table 3: Performance of variants of BERT with number of model parameters measured in millions.

According to our experimental results, we find that the BERT-Large model can generally achieve a higher correction rate than other smaller model by training more. However, the best correction rate achieved by mBERT shows that multilingual BERT models indeed provides significant help with the downstream task involved with low resource languages by extracting the similarity between high resourced languages and the low resourced languages in the pre-training stage. The mBERT model is pre-trained on the corpus of more than one hundred languages. In spite of the considerable number of languages, our experiment result did not suffer from the curse of multilinguality. This is because we treated the misspelled English sentence as a new rare foreign language and turned the misspelling correction task into a translation task, i.e. translating this new and unknown language into the correct English sentence. mBERT is effective at capturing such underlying similarity between this new language and other languages and produces its corrected version.

4.5 Model Concatenation

We incorporate a subword BERT into an encoder-decoder (EncDec) model for spell correction (Kaneko et al., 2020). The concatenated models have two types of tokenizations: subword representation and the other depends on the candidate network. We use the output of fine-tuned BERT as additional features into the spell correction model in two ways, take Char-CNN-LSTM as an example:

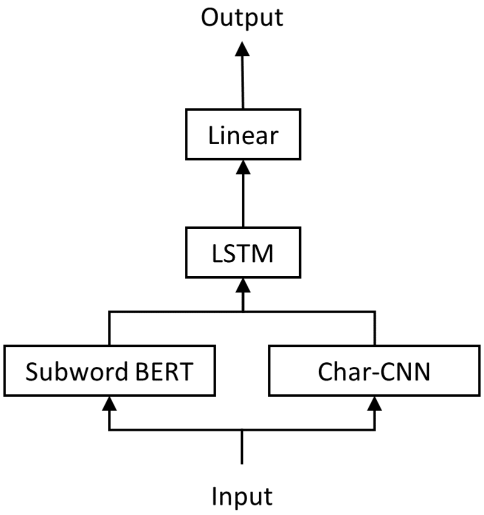

Figure 3: Subword BERT connected to the input.

- **BERT connected at input:** BERT is paralleled with a Char-CNN to form an encoder, and their outputs are then concatenated and fed into an LSTM network. The final prediction is produced by going through an additional linear layer.

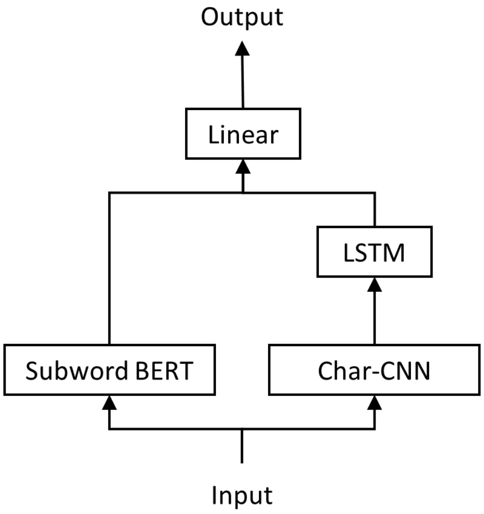

Figure 4: Subword BERT connected to the output.

- **BERT connected at output:**. We employ BERT to extract features from input, paralleled with a whole candidate network, here Char-CNN-LSTM. Their outputs are then concatenated and used to predict the same way as in the previous approach.

Three networks in 4.2 are considered to be the candidate networks, see Figure 3, 4 as illustrations.

Model	**CR (%)**
Char-CNN-LSTM + BERT (output)	**68.80**
Char-CNN-LSTM + BERT (input)	67.92
Char-LSTM-LSTM + BERT (output)	67.45
Char-LSTM-LSTM + BERT (input)	62.63
SC-LSTM + BERT (output)	68.10
SC-LSTM + BERT (input)	68.37
BERT	68.67

Table 4: Performance of concatenated models.

From Table 4 we found that the sub-word BERT model consistently outperforms most of the augmented models, though outranked by Char-CNN-LSTM + BERT (output). It is worth mentioning that the correction rate of Char-LSTM-LSTM + BERT (input) is only 62.63%, meaning that Char-LSTM-LSTM does not fit into the first approach with subword BERT.

We observe that when we employed BERT and augmented it with various candidate networks, the correction rates would not necessarily be better. This is reasonable in the way that the information processed by the candidate network is insufficient compared to representations learned by BERT. That is, the candidate network learned worse than BERT and the concatenation does not help. One exception is the Char-CNN-LSTM + BERT (output) model, character-level CNN-LSTM has shown to be capable of efficiently recognising noisy patterns in spell correction task (Ghosh and Kristensson, 2017). One way to amend the concatenated models is to employ the current structure as the encoder and add a separate decoder as a correction network. Other solutions are possible with deeper candidate networks of which representations are different from what BERT learns.

4.6 CLMBER

We propose our new model CLMBER: Char-**C**NN-**L**STM-**M**ultilingual-**BER**T. A pre-trained mBERT model with its last 3 layers fine-tuned is connected to the output and concatenated with a Char-CNN-LSTM network. The correction rate is 73.39% on test data, which outperforms all models above.

In table 5 we present several examples where CLMBER corrects noisy sentences with both nat-

Ground Truth	Noisy input	Prediction
He was an old man who fished alone in a skiff in the Gulf Stream and he had gone eighty-four days now without taking a fish.[1]	He was an old man who fished **alne** in a skiff in the Gulf **straem** and he **ahd** gone eighty-four days now **whtiour takifng** a fish.	He was an old man who fished **alone** in a skiff in the Gulf **stream** and he **had** gone eighty-four days now **without taking** a fish.
In my younger and more vulnerable years my father gave me some advice that I have been turning over in my mind ever since.[2]	In my younger **ad** more **vulnaerable** years my **fajter gabe** me **ome** advice that I **habe beeen turnening** over in my mind ever since.	In my younger **and** more **vulnerable** years my **father gave** me **some** advice that I **have been turning** over in my mind ever since.
He had just entered, wearing an embroidered court uniform, knee breeches, and shoes, and had stars on his breast and a serene expression on his flat face.[3]	he **ha juss** entered, wearing **a embroodered** court uniform, **bnee** breeches, and **shoies**, and had stars on his **breakst** and a serene **expresion** on his flat face.	he **had just** entered, wearing **a embroidered** court uniform, **knee breaches**, and **shoes**, and had stars on his **breasts** and a serene **expression** on his flat face.

Table 5: Examples of using CLMBER as spell corrector. Green output indicates correct prediction and red output indicates wrong prediction.

Model	CR (%)
CLMBER (fine-tune the last 3 layers)	**73.39**
Char-CNN-LSTM + BERT (output)	68.80
BERT	68.67

Table 6: Performance of CLMBER trained by fine-tuning the last 3 layers.

ural and synthetic errors. In the first[1] and second[2] example sentences, CLMBER predicts the singleton and consecutive spell errors as their correct version. We observed that all misspelled words are corrected, meaning that CLMBER is robust in detecting errors in the sense that wrong words are processed with both subword and character representations though mBERT and Char-CNN-LSTM separately. In the last sentence[3], “a” is copied and not changed to “an”, and “breast” becomes “breasts”, showing that CLMBER can be improved by being further trained on Grammatical Error Datasets.

5 Future work

It is promising to substitute mBERT with other multilingual language models and explore the upper bound of the number of pre-training languages in spell correction tasks. Moreover, since our model is based on multilingual BERT, it is viable to apply it to the spell correction of other languages as long as we modify the tokenization method. Finally, the spell corrector can be further improved via adversarial training, using a generator-discriminator framework (Raheja and Alikaniotis, 2020). Thus, we welcome more experiments in other languages using CLMBER. We hope our work can be helpful to future researches on spell correction task and the analysis of other pre-trained MLMs.

6 Conclusion

The prevailing of the Transformer architecture and BERT models has led to many state-of-the-art results on many NLP tasks. This paper explored the efficiency of pre-trained MLM on the downstream task of English spell correction. Comparisons are made on simple BERT and traditional neural network models. A series of experiments have been made to explore factors influencing the performance of BERT on the spell correction task, including the number of fine-tuned layers, model size, pre-training resources, and model concatenations. Based on our findings, we proposed a new model CLMBER: Char-**C**NN-**L**STM-**M**ultilingual-**BER**T, which leverages all influential factors in our experiments. This new model achieved an improvement of around 5% in the word correction rate, which reinforces our beliefs on the impact of the above factors.

[1] The Old Man and the Sea, Ernest Hemingway, 1951

[2] The Great Gatsby, F. Scott Fitzgerald, 1925

[3] War and Peace, Leo Tolstoy, 1869

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *arXiv preprint arXiv:1904.08398*.

Dimitris Alikaniotis, Vipul Raheja, and Joel R. Tetreault. 2019. The unreasonable effectiveness of transformer language models in grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2019, Florence, Italy, August 2, 2019*, pages 127–133. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2635–2639. ISCA.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

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.

Bohdan Didenko and Julia Shaptala. 2019. Multi-headed architecture based on BERT for grammatical errors correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 246–251, Florence, Italy. Association for Computational Linguistics.

Lluis Formiga and José AR Fonollosa. 2012. Dealing with input noise in statistical machine translation. In *Proceedings of COLING 2012: Posters*, pages 319–328.

Shaona Ghosh and Per Ola Kristensson. 2017. Neural networks for text correction and completion in keyboard decoding. *arXiv preprint arXiv:1709.06429*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Aminul Islam and Diana Inkpen. 2009. Real-word spelling correction using google web 1t 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. Neuspell: A neural spelling correction toolkit. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 158–164. Association for Computational Linguistics.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4248–4254. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-aware neural language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *Acm Computing Surveys (CSUR)*, 24(4):377–439.

Hao Li, Yang Wang, Xinyu Liu, Zhichao Sheng, and Si Wei. 2018. Spelling error correction using a nested rnn model and pseudo training data. *arXiv preprint arXiv:1811.00238*.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Siyuan Qiu, Binxia Xu, Jie Zhang, Yafang Wang, Xiaoyu Shen, Gerard de Melo, Chong Long, and Xiaolong Li. 2020. Easyaug: An automatic textual data augmentation platform for classification tasks. In *Companion Proceedings of the Web Conference 2020*, pages 249–252.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vipul Raheja and Dimitris Alikaniotis. 2020. Adversarial Grammatical Error Correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3075–3087, Online. Association for Computational Linguistics.

Prajit Ramachandran, Peter J. Liu, and Quoc V. Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 383–391. Association for Computational Linguistics.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. Velocitap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 659–668.

Lihao Wang and Xiaoqing Zheng. 2020. Improving grammatical error correction models with purpose-built adversarial examples. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2858–2869.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

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.

Yingbo Zhou, Utkarsh Porwal, and Roberto Konow. 2019. Spelling correction as a foreign language. In *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42st International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019*, volume 2410 of *CEUR Workshop Proceedings*. CEUR-WS.org.