Chinese Grammatical Error Correction on Small Language Models

Long-Hin Fung R12922017 **Henry Ngieng** B10902100 **Szu-Ju Chen** B10705005 **Yen-Chieh Lee** B10902113 **Yan-Tsung Wang** B10902104
Department of Computer Science and Information Engineering
National Taiwan University
{r12922017,b10902100,b10705005,b10902113,b10902104}@ntu.edu.tw

Abstract

In this project, we aim to compare the performance of finetuning smaller models on the task of Chinese Grammatical Error Correction (CGEC), we finetune two pretrained models, bart-base-chinese and mT5, and finetune a qlora module for Taiwan-LLaMa, using the FCGEC and NLPCC datasets seperately. The datasets are collected from examinations of students and news aggregation sites, and sentences written by foreign college students, we evaluate them with two metrics: Exact match, and M^2 scorer, we also submit the predictions of different models to the FCGEC competition for comparison with others.

1 Introduction

In the field of natural language processing, especially since the rise of Large Language Models, people have being relied more and more on this technology. However, such advance technologies come with a cost. The resources needed to run on a Large Language Model privately is far too much for an average user to afford, so large companies would provide the model as a service so that the users could interact with the model without hosting the model himself. This is not a good solution, for it sacrifices the privacy of the user.

Sometimes the user may not want anyone else to know the input, say the user is writing an important private email and wanted to check the grammatical correctness of his email, so instead of trusting the hosting company, he may prefer to run a model himself.

Although there are other methods to maintain the secrecy of the input without running a model locally, for example, one can encrypt the model and input and ask a cloud computing service provider to finish the computation for you with CRYPTEN (Knott et al., 2021), then you can decrypt the result, but you lose quite a lot of speed in exchange for security, so running a model privately would make more sense in this scenario.

In this project, we focus on the task of CGEC. The goal of this task is to correct grammatical mistakes with small models, which can be run on consumers' grade hardware, and our aim is to compare the performance of various small models on this task. Our codes are available at https://github.com/LH104729/112-1-CSIE5413-ADL-Final-Project.

2 Previous Work

There is a lot of work on the task of English Grammatical Error Correction (GEC). Bryant and Briscoe 2018 proposes a language model based approach. The algorithm iteratively calculates log probability, and selects the sequence with higher probability, which means that the sequence contains less errors. Kaneko et al. 2020 incorporates a pretrained masked language model into and encoder-decoder based model for GEC. Rothe et al. 2022 introduces a language-agnostic pretraining method, leveraging large-scale multilingual models, successfully applied to the task of GEC. The approach achieves unsupervised multilingual GEC learning.

However, there seems to be less work (Zhao et al., 2018; Rao et al., 2020; Zhang et al., 2022) on the task of Chinese Grammatical Error Correction. Fortunately, in this year, Fan et al. 2023 introduces GrammarGPT, a novel model for studying the potential of open-source LLMs architectures in addressing CGEC through supervised fine-tuning.

3 Method

We finetune two pretrained models bart-base-chinese (Shao et al., 2021) and mT5 (Xue et al., 2021), and finetune a qlora module for Taiwan-LLaMa (Lin and Chen, 2023) on the FCGEC dataset (Xu et al., 2022) and the NLPCC18 dataset (Zhao et al., 2018).

3.1 Datasets

For FCGEC, the dataset is mainly sourced from examination questions in elementary, middle, and high school and news aggregation websites. The training set contains 36,341 sentences, and the evaluation set consists of 2,000 samples, both the training and evaluation sets are very balanced.

For NLPCC18, the dataset is sourced from sentences written by foreign college students, containing 717,241 sentences. We use 10,000 of them in the training set, while the evaluation set consists of 5,000 samples, both of whom are unbalanced. There are more grammatically incorrect testcases as a direct result of the data collection method.

Both datasets have their advantages and disadvantages. For FCGEC, one major disadvantage is that the data are very political biased, with the word " 中国" appearing 5,264 times in the 36,341 sentences of the training dataset, and one minor disadvantage is the lack of documentation on the format of the dataset. The training and validation dataset have different formatting rules. Fortunately, the content quality of the dataset is really good, covering a lot of native Chinese speakers' subtle grammar mistakes. As for NLPCC18, one obvious advantage is that it covers quite a lot of grammar mistakes, including rare errors occurring in native speakers. However, the meaning of the sentences and the correction for the sentences are quite poor compared to FCGEC, and sometimes the meaning of the input sentences and the model answers are different.

3.2 Data Preprocessing

We prepared 3 data sets for training. The first two are obtained by taking 10,000 samples for each dataset, respectively, and the third is the whole FCGEC training data set. We take 10,000 samples because it is a reasonable size for us to train, and the third set is for submitting to the FCGEC competition. Since there are a lot of ways to correct an ill-formed sentence, there are a lot of answers for each sentence. Hence we take the first answer in the training stage of each sample. As for the validation stage, we compare the model generated answer with all available answers and take the maximum of those scores.

Also there are some weird characters which cause problems for us, so we replaced them, in particular, the character representing three dots with three normal dots, and full-width dashes with a half-width dashes.

Finally, since the data of FCGEC are in simplified Chinese, we translated them to traditional Chinese with OpenCC [1].

3.3 Metrics

It is quite hard to come up with a suitable metric for this problem, as sometimes the corrections are subtle. Hence the score may be similar for correcting the input sentence and outputting the input sentence without modification.

Exact match This is the most Naïve metric as it simply gives a score when the generated correction is exactly the same as the model answer. This clearly avoids giving high score to the model which doesn't correct the sentence at all. Note that one sentence may have a lot of different corrections, so we calculate a prediction as a match if it matches any of the correct answers.

M^2 Scorer *MaxMatch* (M^2) is a widely used algorithm for evaluating grammatical error correction. It computes the phrase-level edits between the source sentence and the output sentence. Then, we compute the precision, recall and $F_{0.5}$-score based on the edit set of our models' outputs and the gold (correct) edit set. Suppose the edit set of our models' outputs is $\{e_1, e_2, ..., e_n\}$ and the gold edit set is $\{g_1, g_2, ..., g_n\}$, the precision, recall and $F_{0.5}$ are defined as follow:

$$P = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |e_i|} \quad (1)$$

$$R = \frac{\sum_{i=1}^{n} |e_i \cap g_i|}{\sum_{i=1}^{n} |g_i|} \quad (2)$$

[1] https://github.com/BYVoid/OpenCC

$$F_{0.5} = 5 \times \frac{P \times R}{P + 4 \times R} \quad (3)$$

Where

$$e_i \cap g_i = \{e \in e_i \mid \exists g \in g_i \text{ s.t. } g = e\} \quad (4)$$

For example, given the following source sentence:

中學生寫作文，要留心觀查各種事物、各種現象，要有真情實感，切忌不要胡編亂造。

If the model edits and the gold edits are:

Model edits (e_i): {觀查 → 觀察}
Gold edits (g_i): {觀查 → 觀察, 不要 → ϵ}

The corresponding $P = 1, R = 0.5, F_{0.5} = 0.8\bar{3}$.

4 Experiments

4.1 BART

We finetune the fnlp/bart-base-chinese(Shao et al., 2021) model with voidful/bart-base-chinese's tokenizer. Since the tokenizer used in fnlp/bart-base-chinese is based on BERT tokens, and BART no longer supports BERT's tokenizer, we are now using voidful/bart-base-chinese's tokenizer. We train the model with following parameters: max source length = 512, batch size = 4, accumulation steps = 1, learning rate = 10^{-4}, num train epochs = 24. Then, we inference the result of each checkpoint and select the best checkpoint.

The Figure 1 shown below is the training loss curve of the fine-tuning process on BART. As you can see that the loss value are still about 3.5 after 14 epochs which is quite big. This might be related to the poor performance on our result.

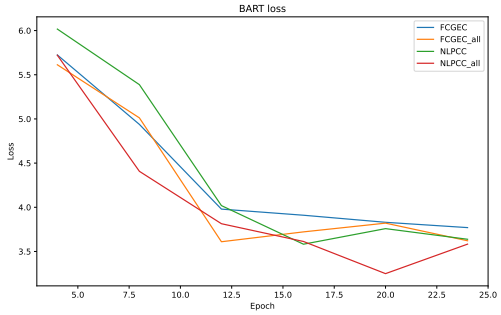

Figure 1: Training loss curves on BART

4.2 mT5

We finetune the google/mt5-small model with the MT5tokenizer and the MT5ForConditionalGeneration load pre-trained method. Here are the hyper parameters we used: learning rate = 1×10^{-4}, training epochs = 15, max source length = 512, max target length = 512, warmup steps = 300, batch size = 2, and gradient accumulation steps = 2. Also, since there could be a source prefix for the mT5 model to keep the training on the same task, we add the source prefix as " 修正錯誤" to stick the training on the CGEC. After the training, we inference the result with source/target length = 512, which is same as the training process and beam = 5 to choose between multiple possible prediction.

After observing the prediction from the fine-tuned model, we have found out that most of the punctuation are in halfwidth form. However, we know that all the Chinese characters and the punctuation are in fullwidth. We have done some postprocess on the result to change those punctuation including comma, exclamation mark, question mark, semicolon, and colon into desired format.

The Figure 2 shown below is the training loss curve of the fine-tuning process on mT5. We can easily observe the learning curve has significantly drop down after two epochs. Also, the learning curve has also gone steady as the epoch grows. It even results in a loss = 0.095 on the large FCGEC dataset.

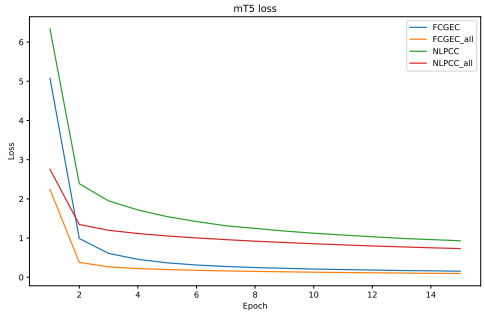

Figure 2: Training loss curves on mT5

4.3 Taiwan LLaMa

We trained a qlora module for Taiwan LLaMa using axoltol(axolotl, 2023) with the following parameters: lora r = 16, lora α = 32, lora

dropout = 0.1, targeting all linear modules, with a cosine dynamic learning rate starting with 0.0002, we trained 3 epochs with batch size of 2×4(gradient accumulation steps), we also do 4 bit quantization and use bf16 in the training process. We used the following prompt format for training and evaluation of Taiwan LLaMa, " 你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。USER: 請修正文法錯誤：{Input} ASSISTANT:{Answer}".

The Figure 3 shown below is the training loss curve of the fine-tuning process on Taiwan-LLaMa. We can see that there always has a big drop on the train loss after each epoch. Also, its final training loss on all four datasets are much smaller than the other two models. It has a nearly 0.02 on FCGEC dataset and a 0.14 on NLPCC dataset.

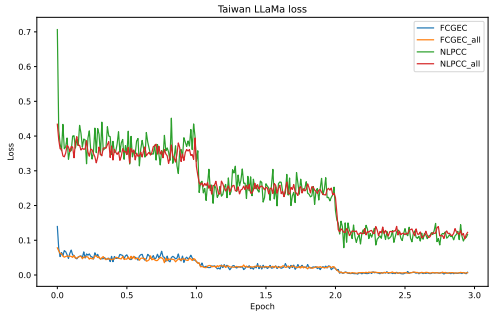

Figure 3: Training loss curves on TWLLM

We repeat the training process on above mentioned three models for the following four training sets: 10,000 samples from FCGEC, all samples from FCGEC, 10,000 samples from NLPCC2018, and all samples from NLPCC2018. Then we evaluate the first two on the 2,000 sample FCGEC validation set, and the last one on the 5,000 sample NLPCC2018 validation set.

4.4 FCGEC competition

We have also submitted some models to the FCGEC competition[2] to see how they perform compared with others. The competition gives us a private testcase to predict on and send the predictions to them. They have six metric for each submission: Binary accuracy, Binary $F1$ score, Type accuracy, Type $F1$ score, Correction exact match, and Correction $F_{0.5}$. the first two are just accuracy and $F1$ score of predicts whether the sentence is grammatically correct or not. The next two are the type of the grammatical errors, while the last two are the exact match score and $F_{0.5}$ score from the M^2 metric. We transform our prediction into the format of the competition by predicting every grammatical error is of some constant type, and disregard the type scores since the error type is not the main focus of this project.

[2] https://codalab.lisn.upsaclay.fr/competitions/8020

5 Results

After inspecting the predictions manually, we observe that our model actually modifies it to a grammatically correct version for some of the sentences which are grammatically incorrect, but this version is not the one of the model's answers. The reason is that there are quite a lot of ways for one to correct a sentence, again having a good metric and dataset for this task is very hard.

Regardless of the not one hundred percent accurate assessment, we will still discuss the performance and the difference between these three models by using the metrics we mentioned in Section 3.3.

The exact match scores are as in Table 1, the M^2 scores are as in Table 2, and the FCGEC competition result scores are in Table 3.

5.1 Exact Match Score

In this section, we focus on the Table 1.

In all cases, we can see that BART has a score of zero. After inspecting the output of BART, we see that BART basically rewrite every sentence. It deletes some of the phrases, changes the order of the phrases, removes the punctuation, and adds some unrecognizable symbols and unrelated words. This will never (in our testcases) get an exact match score from the answer.

For the other two models, mT5 and Taiwan-LLaMa are neck and neck where mT5 performs better on FCGEC, while Taiwan-LLaMa performs better on NLPCC on a small training dataset. However, if we give them more training data on the FCGEC case or NLPCC case, Taiwan-LLaMa all performs better than mT5.

Model	Training dataset	EM	EM_{cor}	EM_{incor}	No modification
Taiwan LLaMa (qlora)	FCGEC	0.4420	**0.9633**	0.0164	0.9428
mT5	FCGEC	**0.4625**	0.9299	**0.0808**	**0.8193**
BART	FCGEC	0	0	0	0
Taiwan LLaMa (qlora)	FCGEC (all)	**0.6035**	**0.9655**	**0.3079**	**0.4768**
mT5	FCGEC (all)	0.5215	0.9199	0.1962	0.5976
BART	FCGEC (all)	0	0	0	0
Taiwan LLaMa (qlora)	NLPCC	**0.1754**	0.7717	**0.0426**	**0.6266**
mT5	NLPCC	0.1658	**0.8463**	0.0142	0.7645
BART	NLPCC	0	0	0	0
Taiwan LLaMa (qlora)	NLPCC (all)	**0.2180**	0.8134	**0.0854**	**0.4722**
mT5	NLPCC (all)	0.1862	**0.8507**	0.0382	0.6884
BART	NLPCC (all)	0	0	0	0

Table 1: Exact match scores: EM is the percentage of exact matches. EM_{cor} and EM_{incor} are the percentage of exact match over the grammatically correct and incorrect instances, respectively. No modification is the percentage of the incorrect instance which the model did not correct at all.

Model	Training dataset	TP	FP	FN	Precision	Recall	$F_{0.5}$
Taiwan LLaMa (qlora)	FCGEC	51	**91**	1266	0.3592	0.0387	0.1353
mT5	FCGEC	**140**	227	**1181**	**0.3815**	**0.1060**	**0.2510**
BART	FCGEC	53	10531	1257	0.0050	0.0450	0.0061
Taiwan LLaMa (qlora)	FCGEC (all)	**438**	**275**	**879**	**0.6143**	**0.3326**	**0.5253**
mT5	FCGEC (all)	322	376	1002	0.4613	0.2432	0.3912
BART	FCGEC (all)	86	10143	1226	0.0084	0.0655	0.0102
Taiwan LLaMa (qlora)	NLPCC	**574**	1651	**7431**	**0.2580**	**0.0717**	**0.1698**
mT5	NLPCC	223	**1097**	7626	0.1689	0.0284	0.0849
BART	NLPCC	147	19367	7713	0.0075	0.0187	0.0086
Taiwan LLaMa (qlora)	NLPCC (all)	**1108**	2354	**7118**	**0.3200**	**0.1347**	**0.2510**
mT5	NLPCC (all)	483	**1248**	7421	0.2790	0.0611	0.1629
BART	NLPCC (all)	289	19180	7647	0.0148	0.0364	0.0168

Table 2: M^2 scores: Positive is the presence of an edit while negative is the absence of an edit. TP is true positive, FP is false positive, and FN is false negative. Presicion, Recall and $F_{0.5}$ are defined in M^2 Scorer section.

Model	Binary accuracy	Binary $F1$	Correction EM	Correction $F_{0.5}$
Taiwan LLaMa (qlora)	**0.6920** (9)	**0.6820** (9)	**0.3004** (9)	**0.5319** (4)
mT5	0.6040 (11)	0.5840 (11)	0.2346 (18)	0.5128 (5)
Best Binary accuracy	**0.7667 (1)**	0.7612 (2)	0.2630 (11)	0.4116 (11)
Best Binary $F1$	0.7663 (2)	**0.7659 (1)**	0.0000 (25)	0.0000 (25)
Best Correction	0.7497 (3)	0.7490 (3)	**0.4075 (1)**	**0.5865 (1)**

Table 3: FCGEC competition, we have also included the best performing models of each metric in this table, the rankings of our submissions are calculated on 26 Dec, 2023.

5.2 M^2 Scores

In this section, we take a look at the Table 2.

Again, we see that BART performs very bad, its false positive is more than other models by a large margin because it rewrite every sentence with some undesired operations. However, we can see that its score is no longer zero as the last exact match metric. It is because M^2 is less strict than exact match, and the detailed reasons is illustrated in Section 3.3.

Then similar to exact match, mT5 performs better than Taiwam-LLaMa on FCGEC, while Taiwan LLaMa performs better than mT5 on NLPCC in the small training set. Taiwan-LLaMa outperforms mT5 by quite a lot if we give them more data in the M^2 scorer.

5.3 FCGEC Competition

For the competition, our submissions are surprisingly good, as we can see in Table 3. We got forth place out of 25 teams in term of $F_{0.5}$ score.

Note that the type accuracy and the type $F1$ score we mention before are omitted since predicting the error type is not our main focus.

5.4 Model Discussion

After observing the metric results, we then take a deeper look at the three models and discuss the reason on their performance.

BART BART is a model that is a sort of a combination of BERT and GPT. This combination solves each shortage which BERT is not good at sequence generation and GPT is not good at downstream tasks that require knowledge from the whole sequence. This is the reason why we choose BART to be one of our fine-tuning target. However, we find out that BART in our training task doesn't perform as expected by observing the prediction. There are many unrecognizable symbols and reordering of sentences which are not the desired output we want in the grammar correction task. This makes BART perform poorly in our evaluation. We wonder that the BART architecture of reconstructing the input data could be the cause.

mT5 mT5 is a text-to-text generator and has been finetuned on over 101 languages. That is how it can handle the Chinese grammar correction task. Since it is only a text-to-text generator, it can actually be trained to handle multiple different task in the same training dataset. However, we aim at the grammar correction only so we have added the " 修正錯誤" prefix in the training process to keep it on a single task. We think that this prefix helps the model to understand that it is working on only one task quickly. Also, the FCGEC dataset is focused more on political based which narrowed down its topic into a much simpler area and have common grammar mistakes that the mT5 pretrained datasets may covered. These are the two main factors we suggest that mT5 has better performance in a small FCGEC dataset.

Taiwan LLaMa Taiwan-LLaMa is a LLaMa-based language model. It is totally pretrained on the traditional Chinese dataset. Moreover, it has a instruction-tuning feature. First, it can understand and learn more on the rare grammar mistakes happened in NLPCC dataset. Second, it has a stronger understanding of the instruction of fixing grammar mistake after more training data given in large FCGEC dataset since the prefix in mT5 only helps to stick the training on the same task. Third, LLaMa is a model that will generate the output based on the last output which may eliminate the possible grammar mistakes. We believe these are the reasons why Taiwan-LLaMa outperforms the other two models in the large FCGEC and NLPCC dataset.

6 Future work

There are quite a few different directions. First, because the data set that we use is not too ideal, we will like to create a data set which is less political, and also contains instances from non native speakers. Second, we will also want to compare our performance to Large Language Models like GPT3 and GPT4. However, due to query limit and money constraints, we are not able to complete such task in this project. Third, since Taiwan-LLaMa performs so well, we would also want to try other methods to finetune Taiwan-LLaMa and see if it can perform better.

7 Conclusion

We finetuned two pretrained models, bart-base-chinese and mT5, and finetuned a qlora module for TaiwanLLaMa on the task of CGEC using the FCGEC and NLPCC datasets seperately. In the experiments, we see that BART performs very badly in all cases, while mT5 and Taiwan-LLaMa are neck and neck on a small training set, Taiwan LLaMa outperforms mT5 with a larger training set, and Taiwan-LLaMa also performs very well in the FCGEC competition. We would like to conclude that the Taiwan LLaMa has a great potential in learning some rare grammar mistakes and details from a large training datset. Thus, we can try other finetuning methods for Taiwan-LLaMa for achieving

better results in the future. Other than the model's result, we have also discovered that the currently available dataset are not ideal, and perhaps we can create a better dataset in the future to obtain a better performance.

References

axolotl. 2023. OpenAccess-AI-Collective/axolotl. https://github.com/OpenAccess-AI-Collective/axolotl.

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253, New Orleans, Louisiana. Association for Computational Linguistics.

Yaxin Fan, Feng Jiang, Peifeng Li, and Haizhou Li. 2023. Grammargpt: Exploring open-source llms for native chinese grammatical error correction with supervised fine-tuning.

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.

B. Knott, S. Venkataraman, A.Y. Hannun, S. Sengupta, M. Ibrahim, and L.J.P. van der Maaten. 2021. Crypten: Secure multi-party computation meets machine learning. In *arXiv 2109.00984*.

Yen-Ting Lin and Yun-Nung Chen. 2023. Taiwan llm: Bridging the linguistic divide with a culturally aligned language model.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2022. A simple recipe for multilingual grammatical error correction.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. FCGEC: Fine-grained corpus for Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing*.