CCL2022-CLTC赛道四：棒棒冰队评测报告

刘旺旺
视源电子科技有限公司/ 广州
liuwangwang@cvte.com

丘文波
视源电子科技有限公司/ 广州
qiuwenbo@cvte.com

摘要

本文档描述了我们在CCL2022-CLTC赛道四，多参考多来源汉语学习者文本纠错任务中提交的参赛系统。该任务主要针对汉语学习者所产生的文本错误进行检测并纠正。评测数据包含不同来源的文本，其蕴含的错误类型存在一定的差异，因此，处理起来有一定的难度。我们的参赛系统主要选择了与评测数据同来源的数据，进行了两阶段训练，并且以投票的方式集成了多个seq2seq与seq2edit模型。最后，在评测数据集上，纠错F0.5达到了50.17，排名第三。在长期榜单，纠错F0.5达到了**52.16**。

关键词： 文本纠错 ； 预训练模型 ； 多阶段训练

CCL2022-CLTC Track 4: Lollipop Team Report

Wangwang Liu
CVTE / Guangzhou
liuwangwang@cvte.com

Wenbo Qiu
CVTE / Guangzhou
qiuwenbo@cvte.com

Abstract

The document describes our submission system in the CCL2022-CLTC Track 4, Multi-Reference Multi-Source Chinese Learner Text Correction Task.This task is mainly aimed at detecting and correcting text errors generated by Chinese learners. Evaluation data of the task contains multi different sources, and the types of errors are vary, so it is difficult to handle. Our system selected data from the same source as the train data, conducted two-stage training, and integrated multiple seq2seq and seq2edit models by voting. Finally, on the evaluation data , the F0.5 of our system reached 50.17, ranking third. In the long-term list, the F0.5 has reached 52.16.

Keywords: Text errror Correction , Pre-training model , Multi-stage training

1 引言

文本纠错任务，即对输入文本中所包含的拼写错误，语法错误，语义错误等进行检测和纠正。它对教育，出版，搜索引擎等领域都有着至关重要的作用。这些错误不仅会影响阅读，而

且可能完全改变文本传递的意义。同时，也会影响分词，词性标注，命名实体识别等NLP基础任务的性能。

我们的参赛系统，首先用所有能获取到的纠错数据进行第一阶段的预训练，然后选择了与评测数据同来源的数据进行了微调，并且以投票的方式集成了多个seq2seq与seq2edit模型。此外，还利用拼写纠错模型，统计语言模型等进行后处理。我们的系统主要参考比赛给出的baseline代码[0]以及其中提到的一些提升方法。

2 数据分析

该任务对汉语学习者所产生的文本错误，进行检测并纠正。给定输入句子，输出纠正后的句子。如Table 1所示，该句有四处修改，两个插入，一个替换，一个删除。我们不需要提交具体的错误类型或者错误位置，只需要提交模型修改后的结果。如果该句没有错误，则输出“106 还有想作为好老师的话，不能给孩子一个答案。还有想作为好老师的话，不能给孩子一个答案。”即可。

输入	106 还有想作为好老师的话，不能给孩子一个答案。
输出	106 还有想作为好老师的话，不能给孩子一个答案。 想做一名好老师的话，就不能只给孩子一个答案。
修改	位置0-2，删除**还有** 位置3-5，替换**作为**为**做一名** 位置11，插入**就** 位置13，插入**只**

Table 1: 文本纠错任务示例

比赛分别给出了1137句有标注验证集以及6000句无标注的测试集。数据来源于lang8，hsk与cged并进行了重新标注，具体详见Zhang et al. (2022)。测试集最小长度2，最大长度257，平均长度37.4。验证集分布情况见Table 2。从表中可以看出，有错误句子数占比很高，并且错误句子所包含的错误个数也比较多。

句子数	1137
最长句子字数	395
最短句子字数	10
字数/句子数	75.4
有错误句子数占比	0.952
错误数/ 错误句子数	3.75
错误字数/ 错误数	1.53

Table 2: 验证集分布情况

使用ChERRANT[1]工具，对验证集的错误进行统计分析，错误类型分布以及错误词性分布如下图 1，图 2。从图中可看出，按照错误类型划分，误用错误即替换错误占比最多，超过半数，其次是缺失错误，占比约为28%。按照错误词性划分，动词错误占比最高，约为21%，其次是标点错误。其他错误一般包含多个错误的修改，可能同时涉及标点，名词，动词等。更具体详细的评测数据以及任务相关信息，可参考CCL2022-CLTC评测概述Wang et al. (2022)。

3 系统

3.1 模型

我们使用的模型结构与比赛给出的baseline基本一致。

[0]https://github.com/HillZhang1999/MuCGEC
[1]https://github.com/HillZhang1999/MuCGEC/tree/main/scorers/ChERRANT

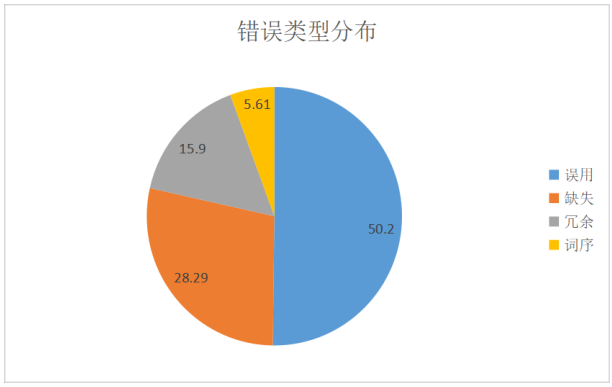

Figure 1: 错误类型分布

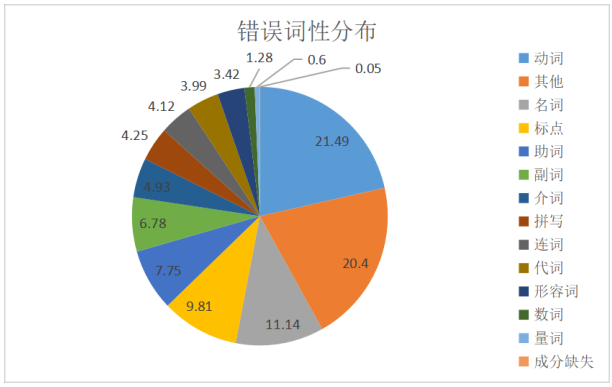

Figure 2: 错误词性分布

（1）seq2edit模型：使用bert Kenton and Toutanova (2019)系列的预训练模型作为编码器，训练gector Omelianchuk et al. (2020)模型。首先，获得输入文本中的每个字符的向量表示，然后，通过全连接层，预测每个位置对应的类别，每个类别代表一种编辑（图 3仅仅给出标黄字符对应的编辑）。推理阶段，根据输入文本及预测到的每个字符对应的编辑标签，即可得到正确的句子。为了能够修改一些连续错误，将上述过程迭代多次，直至不产生新的修改。

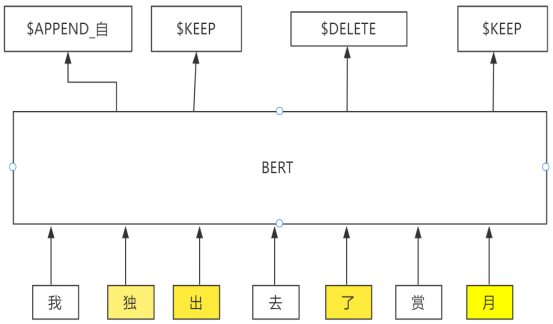

Figure 3: seq2edit模型

（2）seq2seq模型：基于BART Shao et al. (2021)预训练模型，训练seq2seq模型，输入为错误句子，输出为正确句子。推理阶段，先对输入句子进行编码，然后以自回归的方式逐个预测下一个位置可能的字符，并作为下一个预测的输入，如图 4虚线部分。

3.2 训练数据

我们获取了多个来源的中文纠错通用标注语料：

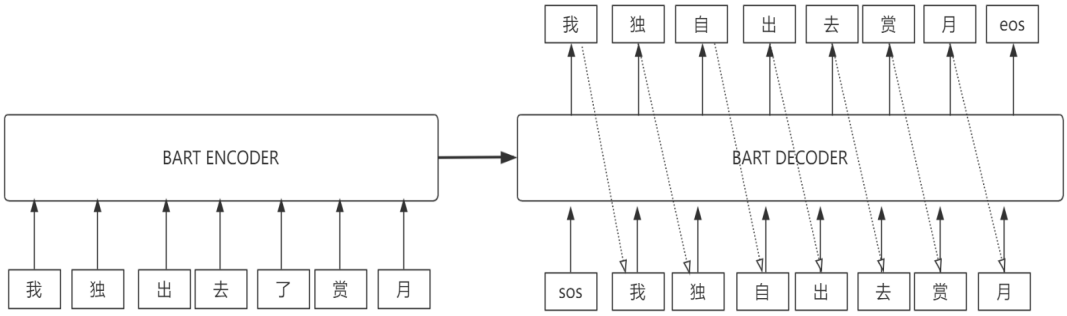

Figure 4: seq2seq模型

（1）lang8[2]：根据Lang-8语言学习网站中母语者对汉语学习者作文的修改记录生成。

（2）hsk：北京语言大学对汉语学习者参加汉语水平考试中写的作文组织人员进行了语病标注。

（3）cged[3]：中文句法错误诊断技术评测自2014-2022所发布的数据。

（4）ctc2021[4]：CTC2021中文文本纠错比赛公开数据集。

（5）wikideits[5]：中文维基百科编辑历史抽取并过滤得到的语料。

所有数据经过去重处理，并且仅仅包含有错误句子对，也严格按照比赛要求进行剔除处理，各数据集统计如下：

通用标注语料	数据量（句）
lang8	1076336
hsk	91175
cged	45245
ctc2021	215852
wikideits	4628049
总计	6056657

Table 3: 训练数据

3.3 模型的集成及后处理

模型集成：模型集成是提升比赛结果的重要方法，通过集成多个各方面表现差异较大的模型，可以大大提升系统的效果。对于深度模型，集成的方法也有多种，例如，投票法，概率平均，checkponit参数平均，评分模型等。在本次评测中，我们将多个seq2seq模型与多个seq2edit模型进行投票集成。具体的，先使用ChERRANT，对每个模型的输入输出句子进行处理，获得输入句子对应的修改，若某个修改的次数超过模型总数的一半，则认为修改正确，反之，则忽略修改。其中，ChERRANT根据输入输出句子获得修改，可以是字级别，也可以是词级别。

模型后处理：通过观察集成模型的输出，我们发现模型对拼写错误修改得不是很好，因此增加了拼写纠错后处理，并通过统计语言模型来衡量是否保留某个字符的拼写纠正结果。

[2]https://github.com/blcuicall/CCL2022-CLTC/tree/main/datasets/track1
[3]https://github.com/blcuicall/cged_datasets
[4]https://github.com/destwang/CTC2021
[5]https://github.com/xueyouluo/wiki-error-extract

4 实验

4.1 训练设置

seq2edit模型的预训练模型使用了large版本的macbert Cui et al. (2020)，roberta Cui et al. (2021)，structbert Wang et al. (2019)，进行了两阶段训练：（1）预训练，使用Table 3所示的全部中文纠错通用标注语料（2）微调，使用hsk与cged语料，并且cged重采样两份, 这也与评测数据分布一致。

seq2seq模型的预训练模型使用了large版本的bart Shao et al. (2021)。seq2seq模型训练速度比较慢，我们使用三个不同的随机种子，仅进行了一个阶段的训练，使用lang8，hsk与cged语料。

所有预训练模型均使用transformers库[6]构建，代码均使用pytorch。

4.2 验证集结果

以基于structbert的seq2edit模型为例，不同阶段验证集上的结果如Table 4所示。其中，Inf.tweaks 是推理阶段两个超参数的调整，additional_confidence是Keep标签额外置信度，min_error_probability是句子级别最小修改阈值，当句子错误的概率小于min_error_probability时，句子不修改，设置这两个参数可以保证更高的P值。在structbert模型上，当additional_confidence设为0.15，min_error_probability设为0.40，表现最佳。

structbert	P	R	F0.5
stage1	51.5	25.13	42.57
+ stage2	48.69	32.24	44.18
+ Inf.tweaks	54.58	28.32	46.04

Table 4: seq2edit-structbert验证集结果

从表中可以看出，对单个模型来说，第二阶段的微调，可以进一步提高模型的召回率。推理阶段的超参数的设置也可以进一步提高模型的效果。但是，实验发现经过超参数调整之后的模型，再集成的结果比不做超参数调整的集成结果要差。可能集成模型考虑多个模型的修改，已经可以获得较高的P值。

模型	P	R	F0.5
seq2edit (structbert)	48.69	32.24	44.18
seq2edit (roberta)	47.2	30.45	42.52
seq2edit (macbert)	45.16	33.6	42.26
seq2seq (seed 2021)	46.26	32.67	42.71
seq2seq (seed 200)	46.4	32.28	42.66
seq2seq (seed 300)	46.94	31.58	42.78
seq2edit x 3 + seq2seq x 3 (word)	66.14	25.94	50.49
seq2edit x 3 + seq2seq x 3 (char)	66.6	27.38	51.77

Table 5: 各个单模型及其集成结果

最终使用的各个单模型及其集成结果如Table 5所示。从表中可看出，seq2edit模型在验证集上的效果：structbert >roberta >macbert. seq2seq模型的结果与roberta，macbert相差不大。最后，以投票集成的方式集成了3个seq2edit模型与3个seq2seq模型，整体的P值大大提高，R值略有下降。另外，基于字的集成模型比基于词的结果更好。

不同的模型在不同的错误类型上的表现也有差异，如Table 6所示，seq2edit模型在缺失和词序错误上面的召回要比seq2seq低，尤其在词序错误上面，这也符合seq2edit模型的设计，修改词序错误需要多次迭代。seq2edit模型在替换类型错误上面，表现更好。最终，基于字的集成模型，在处理缺失错误上表现较差。

[6]https://github.com/huggingface/transformers

seq2edit (roberta)			
类型	P	R	F0.5
缺失	50.65	24.75	41.89
冗余	45.17	40.12	44.06
误用	45.44	32.39	42.05
词序	63.49	17.31	41.41
seq2seq (seed 200)			
类型	P	R	F0.5
缺失	46.75	27.07	40.82
冗余	43.41	40.07	42.7
误用	46.55	33.84	43.29
词序	67.42	25.97	51.11
seq2edit x 3 + seq2seq x 3 (char)			
类型	P	R	F0.5
缺失	66.43	20.46	45.83
冗余	59.1	35.07	51.98
误用	71.09	29.59	55.51
词序	75.58	27.43	55.94

Table 6: 不同错误类型对应的指标

4.3 错误样例

Table 8列出了一些修改样例，我们发现存在很多修改其实是正确的，只是和标注不同。除此之外，还存在长难句修改不好，缺失内容太多修改不好，连续错误修改不好，拼写错误修改不好等问题。

4.4 测试集结果

最终，测试集上的结果如Table 7所示。虽然增加拼写纠错后处理可以提高召回率，但也会引入错误。相比比赛取得的结果F0.5为50.17，主要做得改进就是增加了seq2edit模型的两阶段训练。

模型	P	R	F0.5
seq2edit x 3 + seq2seq x 3	69.04	26.36	52.16
seq2edit x 3 + seq2seq x 3 + csc	67.57	27.23	52.12

Table 7: 测试集结果

5 总结

在本次多参考多来源汉语学习者文本纠错任务中，主要难点在于评测数据包含不同来源的文本，错误分布存在一定的差异，并且训练数据也不够充分。我们的方法主要使用与评测数据同来源的数据进行了两阶段训练，并且集成了多个异构模型。最终，我们的系统F0.5的值为50.17，排名第三，长期榜单F0.5的值为52.16.

从实验的结果，可以看出，我们的系统召回率非常低，这主要原因还是训练数据不足，对错误的覆盖度不够，因为时间的原因，也没有做数据增强。其次，还因为集成模型的方法比较简单，过滤掉了正确的修改，一些新的模型集成方法Qorib et al. (2022)有待进一步尝试。

输入	我也来介绍一下**儿**我的家吧。我家有四口人。
标注	我也来介绍一下**儿**我的家吧。我家有四口人。
预测	我也来介绍一下我的家吧。我家有四口人。
错误原因	标注不正确
输入	我不愿意这么过分**的**喜欢歌手。
标注	我不愿意这么过分**的**喜欢歌手。
预测	我不愿意这么过分**地**喜欢歌手。
错误原因	标注不正确
输入	再见，我**有要**很多功课要做。
标注	再见，我**有**很多功课要做。
预测	再见，我**还有**很多功课要做。
错误原因	修改与标注不同，但也是正确的
输入	死刑这个问题是特别争议性。
标注	死刑这个问题特别**具有**争议性。
预测	死刑这个问题特别**有**争议性。。
错误原因	修改与标注不同，但也是正确的
输入	四年前，我高中毕业后时，我母亲被工作去日本住六个月。 因为我还没决定什么大学我想去，我就跟妈妈一起去东京住。
标注	四年前，我高中毕业后，我母亲**由于**工作去日本住六个月。 因为我还没决定我想去什么大学，我就跟妈妈一起去东京住。
预测	四年前，我高中毕业后，我母亲**因**工作去日本住了六个月。 因为我还没决定我想去什么大学，我就跟妈妈一起去东京住。
错误原因	修改与标注不同，但也是正确的
输入	如果你在肺部或呼吸道有病，其治疗就会**难的**。
标注	如果你在肺部或呼吸道有病，其治疗就会**很难**。
预测	如果你在肺部或呼吸道有病，其治疗就会**很难的**。
错误原因	修改错误，修改之后存在杂糅

Table 8: 错误样例分析

参考文献

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: Tag, not rewrite. *ACL 2020*, page 163.

Muhammad Qorib, Seung-Hoon Na, and Hwee Tou Ng. 2022. Frustratingly easy system combination for grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1964–1974.

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*.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*.

Yingying Wang, Cunliang Kong, Xin Liu, Xuezhi Fang, Yue Zhang, Nianning Liang, Tianshuo Zhou, Tianxin Liao, Liner Yang, Zhenghua Li, Gaoqi Rao, Zhenghao Liu, Chen Li, Erhong Yang, Min Zhang, and Maosong Sun. 2022. Overview of cltc 2022 shared task : Chinese learner text correction.

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. *arXiv preprint arXiv:2204.10994*.