CCL23-Eval任务7赛道一系统报告：Suda &Alibaba 文本纠错系统

蒋浩辰[1]，刘雨萌[1]，周厚全[1]，乔子恒[1]，章波[2]，李辰[2]，李正华[1]，张民[1]
1.苏州大学计算机科学与技术学院，苏州，中国
2.阿里巴巴达摩院，杭州，中国
联系邮箱：hcjiang@stu.suda.edu.cn; zhli13@suda.edu.cn

摘要

本报告描述 Suda &Alibaba 纠错团队在 CCL2023 汉语学习者文本纠错评测任务的赛道一：多维度汉语学习者文本纠错（Multidimensional Chinese Learner Text Correction）中提交的参赛系统。在模型方面，本队伍使用了序列到序列和序列到编辑两种纠错模型。在数据方面，本队伍分别使用基于混淆集构造的伪数据、Lang-8 真实数据以及 YACLC 开发集进行三阶段训练；在开放任务上还额外使用 HSK、CGED 等数据进行训练。本队伍还使用了一系列有效的性能提升技术，包括了基于规则的数据增强，数据清洗，后处理以及模型集成等。除此之外，本队伍还在如何使用 GPT3.5、GPT4 等大模型来辅助中文文本纠错上进行了一些探索，提出了一种可以有效避免大模型过纠问题的方法，并尝试了多种 Prompt。在封闭和开放两个任务上，本队伍在最小改动、流利提升和平均 $F_{0.5}$ 得分上均位列第一。

关键词： 文本纠错 ；序列到序列 ；序列到编辑

CCL23-Eval Task 7 Track 1 System Report: Suda &Alibaba Team Text Error Correction System

Haochen Jiang[1], Yumeng Liu[1], Houquan Zhou[1], Ziheng Qiao[1],
Bo Zhang[2], Chen Li[2], Zhenghua Li[1], Min Zhang[1]
1.School of Computer Science and Technology, Soochow University, Suzhou, China
2.DAMO Academy, Alibaba Group, Hangzhou, China
Contact Email: hcjiang@stu.suda.edu.cn; zhli13@suda.edu.cn

Abstract

The article describes the submission of Suda &Alibaba Error Correction Team for Track 1 of the Multidimensional Chinese Learner Text Correction (CCL2023) evaluation task. In terms of models, we used both sequence-to-sequence and sequence-to-edit correction models. For data, we conducted a three-stage training using pseudo data constructed based on confusion sets, real data from Lang-8, and the development set from YACLC. In the open task, we also utilized additional data such as HSK and CGED for training. We employed a series of effective performance enhancement techniques, including rule-based data augmentation, data cleaning, post-processing, and model ensembling. Moreover, we explored the use of large models such as GPT3.5 and GPT4 to assist Chinese text correction and tried various prompts. In both the closed and open tasks, our team ranked first in minimum edits, fluency improvement, and average F0.5 scores.

Keywords: Text Correction , Sequence-to-sequence , Sequence-to-edit

项目资助：国家自然科学基金（62176173）、江苏高校优势学科建设工程资助项目、阿里巴巴AIR计划项目

1 任务介绍

随着互联网时代的到来，大量信息被快速地产生和传播。然而，由于人们的疏忽、打字错误或语言技能不足，文本中往往存在各种错误。这些错误可能会导致信息不准确，甚至对人造成误导。因此，文本纠错任务的重要性日益凸显。在自然语言处理中，文本纠错任务是指通过自动化的方式来检测和修正文本中的错误，包括拼写错误、语法错误和语义错误等。

1.1 任务定义

本次比赛赛道一的多维度汉语学习者文本纠错 (Multidimensional Chinese Learner Text Correction) 任务，旨在自动检测并修改汉语学习者文本 (Chinses Learner Text) 中的标点、拼写、语法、语义等错误，从而获得符合原意的正确句子。其特点是多维度评价，由于同一个语法错误从不同语法点的角度可被划定为不同的性质和类型 (Wang et al., 2021)，也会因语言使用的场景不同、具体需求不同，存在多种正确的修改方案，针对这个情况，多维度汉语学习者文本纠错任务从最小改动 (Minimal Edit，M) 和流利提升 (Fluency Edit，F) 两个维度对模型结果进行评测。最小改动维度要求尽可能好地维持原句的结构，尽可能少地增删、替换句中的词语，使句子符合汉语语法规则；流利提升维度则进一步要求将句子修改得更为流利和地道，符合汉语母语者的表达习惯。

此外，近一年随着人工智能技术的不断进步和发展，一些优秀的大规模语言模型产品逐渐进入人们的视野，如 OpenAI 的 ChatGPT 和 GPT4 (OpenAI, 2023)、百度的文心一言、清华大学的 ChatGLM 等，能够自动地理解、处理和生成文本，并在自动问答、开放域对话等任务中表现出了优异的性能。这些模型的出现也为文本纠错任务的研究提供了新的机遇，因此本次比赛分别设立封闭，开放任务。封闭任务仅允许使用主办方提供的 Lang-8 数据训练，禁止使用大模型，模型参数量需为 10B 以下；开放任务则可使用所有开源数据，且允许参赛队伍使用包括 ChatGPT、文心一言、ChatGLM 等在内的大模型，通过调整 Prompt 等方式来实现更好的纠错效果。

1.2 任务数据

本次评测针对赛道一提供的数据集，包括供参赛队伍训练模型的训练集，供参赛队伍进行模型调优的开发集，以及评测参赛队伍模型性能的封闭测试数据集。训练集来自 NLPCC2018-GEC (Zhao et al., 2018) 发布的采集自 Lang-8 平台的数据。开发和测试数据来源为汉语学习者文本多维标注数据集 YACLC。

针对赛道一，主办方提供了最小改动和流利提升两个维度的多参考数据集 YACLC-Minimal 和 YACLC-Fluency。其中 YACLC-Minimal 属于最小改动维度，YACLC-Fluency 属于流利提升维度。赛道一的数据集统计信息如表 (1) 所示。

	原句数	参考句数	平均参考句数	有修改的参考句数
YACLC-Minimal-Dev	1,839	15,938	8.67	15,935 (99.98%)
YACLC-Minimal-Test	7,296	42,462	5.82	40,334 (94.99%)
YACLC-Fluency-Dev	1,839	3,332	1.81	3,332 (100%)
YACLC-Fluency-Test	5,515	10,237	1.86	8,604 (84.05%)

Table 1: YACLC 数据集统计

2 模型

比赛中，本队伍使用了序列到序列 (Seq2Seq) 模型和序列到编辑 (Seq2Edit) 两种模型，虽然 Seq2Edit 模型整体性能低于 Seq2Seq 模型，但由于这两者的差异性，在集成时它们能起到很好的互补作用。在封闭任务上，我们使用三阶段训练策略，第一阶段使用基于规则加噪的伪数据预训练，第二阶段使用主办方提供的 Lang-8 数据集微调，第三阶段则使用 YACLC 开发集进行精调。在开放任务上，Seq2Seq 使用 NaSGEC (Zhang et al., 2023) 提供的基于 100M 伪数据预训练的 BART 模型；Seq2Edit 则使用自己构建的 10M 伪数据进行预训练。在此基础上进行三阶段微调训练，首先使用 Lang-8 (Zhao et al., 2018)、HSK (Zhang, 2009)、CGED (Rao et al., 2018; Rao et al., 2020)、MuCGEC-Dev (Zhang et al., 2022b)、NLPCC-2018-Test (Zhao

et al., 2018)[1][2] 进行第一阶段训练，然后去掉 Lang-8 进行第二阶段训练，最后使用 YACLC 开发集进行精调。此外我们初步的实验结果显示，在开放任务上使用单参考[3]数据会让模型性能更好，而封闭任务则没有效果。

2.1 基于序列到序列的语法纠错模型

Seq2Seq 是一种深度学习模型，用于处理输入和输出都是序列数据的任务。它在自然语言处理、机器翻译、语音识别等领域具有广泛的应用。

在基于 Seq2Seq 的语法纠错模型中，编码器负责将输入的原始文本序列编码成一个固定长度的向量表示，其中包含了输入文本的语义和上下文信息。解码器则根据编码器生成的向量表示和已知的纠正文本序列，逐步生成纠正后的文本序列。在生成过程中，解码器会考虑上下文信息和目标纠正序列的条件概率，以生成更准确的纠正文本。

Seq2Seq 模型基本的训练数据为一个由原始句子和正确句子所组成的平行句对，本次比赛我们使用 bart-large-chinese[4] (Lewis et al., 2020; Shao et al., 2021) 的 Fairseq 版本[5]作为 Seq2Seq 的基底预训练语言模型。参照 SynGEC (Zhang et al., 2022c)，我们使用更新后词表来训练模型，补充了词表中缺失的中文引号等内容，来让模型达到更高的性能。

2.2 基于序列到编辑的语法纠错模型

Seq2Edit 是一种用于处理序列编辑任务的深度学习模型。它主要用于解决文本编辑、文本改写、机器翻译等任务，其中输入序列需要经过一系列编辑操作来生成目标序列。

序列到编辑模型与传统的序列到序列模型类似，但在解码器的设计上有所不同。传统的序列到序列模型使用自回归 (Autoregressive) 的方式逐步生成输出序列，而序列到编辑模型引入了编辑操作，通过模拟插入、删除和替换等操作来实现序列的改动。

GECToR (Omelianchuk et al., 2020) 是一种基于序列到编辑的语法纠错模型，旨在解决语法纠错任务中的错误检测和纠正问题。GECToR 模型的核心思想是将语法纠错任务视为将原始文本序列转换为纠正后的文本序列的编辑操作序列。具体而言，模型使用 Transformer (Vaswani et al., 2017) 架构，其中包括编码器和解码器。编码器将输入的原始文本序列编码成上下文感知的表示，而解码器则根据这些表示生成一个编辑操作序列，该序列描述了如何将原始文本转换为纠正后的文本。

标签	描述	个数	总数
@@UNKNOWN@@	未知编辑	1	7515
@@PADDING@@	填充token	1	
$KEEP	保持当前token不变	1	
$DELETE	删除当前token	1	
$APPEND(t)	在当前token后新添一个token t	3779	
$REPLACE(t)	将当前token替换为另一个token t	3728	

Table 2: GECToR 模型的标签类型

GECToR 模型的训练过程包括两个阶段：错误检测和错误纠正。在错误检测阶段，模型通过预测每个 token 的编辑操作标签来确定错误位置以及标签。在错误纠正阶段，模型使用编辑操作序列来生成纠正后的文本。这些编辑操作包括插入、删除和替换等，以修正语法错误。

由于 GECToR 模型需要标签，而训练数据只是错误-正确的平行句对，因此首先需要通过基于最小编辑距离算法的标签抽取方法将输入转换成对应的编辑标签序列，然后再送入模型。本次比赛我们使用 bert-struct-large[6] (Wang et al., 2019) 作为序列到编辑的基底预训练语言模型。值得一提的是，Seq2Edit 模型的编辑词表为训练数据中抽取，不在词表中的编辑操作不会

[1] https://github.com/HillZhang1999/MuCGEC#%E8%AE%AD%E7%BB%83%E6%95%B0%E6%8D%AE
[2] https://github.com/blcuicall/cged_datasets
[3] https://blcuicall.org/CCL2022-CLTC/reports/track4/cltc2022-track4-rank1-ye.pdf
[4] https://huggingface.co/fnlp/bart-large-chinese/tree/v1.0
[5] https://github.com/HillZhang1999/SynGEC
[6] https://github.com/alibaba/AliceMind/tree/main/StructBERT

被预测，但词表过大也会干扰模型预测，我们根据训练数据做了一定调整，最终使用的编辑词表为训练数据中重复超过 5 次的编辑，加上了 YACLC 开发集重复超过 2 次的编辑，最终词表中包含了 7515 种编辑操作，对 YACLC 开发集的编辑基本达到了完全覆盖，同时词表规模也比较均衡。

原始句子	力	行	节	约	，	反	反	对	费
编辑标签	$R(厉)	$K	$K	$K	$K	$K	$D	$A(浪)	$K
纠错结果	厉	行	节	约	，	反		对浪	费

Table 3: 编辑标签抽取示例。(R: REPLACE, K: KEEP, D: DELETE, A: APPEND)

3 性能提升技术

在两种模型的基础上，本队伍也尝试了一些泛用性较强的性能提升技术，包括数据增强，数据清洗，基于规则的后处理以及模型集成四部分，本次比赛我们对于仅拟合单一数据集的技术探索较少。

3.1 基于规则的数据增强

在机器学习和深度学习任务中，模型通常需要大量的训练数据才能取得良好的性能，但实际应用中我们的数据往往是有限的。数据增强是指对已有数据进行一系列变换和扩充，模拟真实数据生成伪数据的技术，在训练数据有限的情况下，合理的数据增强方法能较好地提高模型性能。

本队伍使用了基于规则的数据增强方法 (Zhang et al., 2021)，我们首先对主办方提供的 YACLC 开发集做了分析，统计了不同错误类型的分布，结果如图 (1) 所示，我们模拟该分布，基于混淆集、近义词词表和同音词词表 (Zhang et al., 2022a)，对句子中的字、词以一定概率随机进行替换、插入、删除和词序调换等操作，获得人造伪数据。

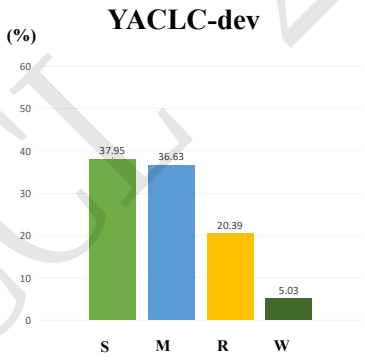

Figure 1: YACLC 最小改动维度开发集的错误分布

具体而言，我们将句子切为等长子段，对每个子段以一定概率随机选取 1-3 个 token 进行加噪。加噪方式分为替换、添加、删除和词序调换四种，每种加噪方式又被分为字和词两种级别。各加噪方式的概率与数据集对应的错误分布相同。

混淆集	原字/词	混淆集内容
同音混淆集（词级别）	不是	不时/捕食/不适/不实...
同音混淆集（字级别）	于	与/育/语/预/域/余/遇...
形近混淆集（字级别）	特	痔/持/待/恃/诗/侍/峙...

Table 4: 混淆集示例

第二十二届中国计算语言学大会论文集，第220页-第229页，哈尔滨，中国，2023年8月3日至5日。

我们在 YACLC 最小改动维度开发集上测试了数据增强的效果，其在 Seq2Edit 模型上能带来 0.5% 的提高，Seq2Seq 模型上的提高为 1% 左右，主要的提升由召回率带来，详细结果如表 (6) 所示。

3.2 Lang-8 数据清洗

数据清洗是指在数据分析和处理过程中，对原始数据进行检查、修正、删除或补充等操作，以确保数据的准确性、完整性和一致性。观察比赛给定的 Lang-8 训练集，我们发现其中含有一定量的噪音数据，如表 (5) 所示。

这些数据原句和目标句的差异过大，显然不属于纠错范畴，对模型的训练会产生干扰。本队伍通过人工规则清洗此类噪音数据。具体而言，去除答案是原句长度 1.5 倍以上的训练数据，对重复导致的过长目标句，保留前面和原句相似的部分，其余无法修正的数据则直接舍弃。最终共修正 20K 噪音数据、筛去了 30K 条噪音数据。

原始句子	渐渐地天气很冷。
目标句子	渐渐地天气变冷了。OR 天气已经很冷了。
原始句子	他的口太小。
目标句子	他的口太小，而且还没有一颗牙齿

Table 5: Lang-8 中的噪音数据

我们同样在 YACLC 最小改动维度开发集上测试了数据清洗技术的效果，数据清洗带来的提升比较稳定，平均提升 1% – 3%，实验结果如表 (6) 所示。

3.3 基于规则的后处理

由于词表不能覆盖数据中的所有词，纠错模型预测时会将不在词表中的 token 用“<unk>”标识符表示，更新词表后“<unk>”标识符极大减少但仍然存在。此外，Seq2Seq 模型在输入和预测时都使用了 BPE 分词，因此最终输出会包含“##” BPE 分词标记，在后处理时，我们将“<unk>”标识符还原，同时基于规则除去了所有 BPE 分词标记。值得注意的是，因为使用的训练数据和预训练模型都基于中文，纠错模型对英文的纠错能力较差，因此我们筛去了所有有关英文的编辑，保留英文原文。

考虑到模型预测的标点中英文混杂，本队伍还尝试了将所有标点转换为中文，但转换后的 $F_{0.5}$ 分值反而会有所下降，观察 YACLC 开发集后我们发现其中对标点的要求并不严格，其并未将标点全改为中文，因此最终我们未对标点做相关处理。

后处理的效果可以参考表 (6)，由于 BPE 分词标记和重写的原因，其在 BART 模型上的提升较大。

3.4 模型集成

模型集成是一种将多个独立的模型组合在一起，以获得更强大和更准确的预测结果的技术。通过结合多个模型的预测，模型集成可以降低单个模型的偏差、方差或错误率，提高整体性能和稳定性。

本队伍使用投票的方式集成不同模型的结果，其优点是直接在预测结果上集成，可以有效应对模型结构多样的场景。具体而言，我们首先将预测结果转为 m2 格式并抽取出编辑，然后在编辑级别进行投票，采纳票数超过投票阈值 k 的编辑，不同模型可以自定义投票权重，若采纳的编辑范围重复，则随机选择其中一种，当集成模型数量较多时，可以通过投票阈值的设置来控制精确率和召回率的平衡。

通常，投票集成的效果受到以下因素影响：首先集成模型的性能需要相近，若性能差距较大，低性能的模型会干扰集成结果，如果性能差距无法避免，则可以优先保证精确率相近，或是调整低性能模型的投票权重来弱化干扰。在此基础上，我们认为集成模型之间的差异性越大越好，以本次比赛使用的 Seq2Edit 和 Seq2Seq 模型为例，它们模型结构不同，擅长纠正的错误类型也有较大差异，因此能起到很好的互补作用，实验显示即使单 Seq2Edit 模型的性能比 Seq2Seq 模型低了 3% – 4%，二者的集成效果也超过了纯 Seq2Seq 模型集成。

模型	最小改动维度		
	P	**R**	**$F_{0.5}$**
Seq2Edit	59.52	35.59	52.46
+pseudo	58.72	38.29	53.06
+clean	59.34	39.03	53.75
+post process	59.66	35.65	52.58
Seq2Seq	67.87	39.28	59.24
+pseudo	67.06	43.12	60.36
+clean	67.57	48.24	62.56
+post process	65.47	46.10	60.39
6×Seq2Seq	72.81	39.64	62.37
3×Seq2Seq+3×Seq2Edit	74.78	42.84	65.07

Table 6: 各性能提升技术实验结果 (pseudo 为使用数据增强技术训练的模型，clean 为使用数据清洗技术训练的模型,post process 为后处理后的结果，实验均在 YACLC 最小改动维度开发集上测试，集成使用的模型为 baseline 模型)

4 大模型辅助纠错

本次比赛设有开放任务，允许使用大模型辅助纠错。本队伍针对中文纠错任务设计了一套简单的 Prompt ，分别使用 GPT3.5 和 GPT4 预测了对应的纠错结果，对利用大模型进行纠错进行了一些探索。

4.1 Prompt 设计

大模型的 Prompt 设计会对结果带来直接影响，设计合理且有效的 Prompt 对使用大模型至关重要，本队伍结合前人设计 Prompt 的经验，多次尝试后构造了一个简洁的语法纠错 Prompt ，其由角色 Prompt、指令 Prompt、输出控制 Prompt 和一致性 Prompt 四部分构成。角色 Prompt 让大模型扮演一个中文领域的语言专家；指令 Prompt 则尽可能以简洁精确的语言描述中文纠错，让大模型理解自己的任务；输出控制 Prompt 用于指定结果输出的格式，便于对预测结果的后处理；一致性 Prompt 则用于强调输入输出的一致性，避免缺漏数据的情况。完整的 Prompt 请参考表 (7)。在上述 Prompt 的基础上，我们随机采样十条开发集的数据作为 example，以输出控制 Prompt 中指定的格式输入模型。我们发现加入 example 能让大模型更好地理解输入和输出，但对纠错性能的提升不大。

值得一提的是，也许是大模型训练数据中英文占比较高的原因，使用英文 Prompt 有助于模型更好地理解任务目标，实验显示英文 Prompt 相较于中文 Prompt 会对结果有较稳定的提升，详细结果可参考表 (10)。由于大模型直接在测试集上评测的 $F_{0.5}$ 值会因过度改写纠正的原因被大幅影响，且难以通过简单的 Prompt 指令避免，因此我们将大模型参与集成后的结果作为评价标准，具体集成方式可参考第 4.3 节，通俗来讲就是不考虑过纠问题，只关注大模型对语法错误的纠正能力。

第二十二届中国计算语言学大会论文集，第220页-第229页，哈尔滨，中国，2023年8月3日至5日。

角色+指令 Prompt	请你以中文领域语言专家的身份，纠正句子中的各种语法错误，使其符合中文的表达习惯但不改变句子原意。在修改错误时，你首先需要理解整个句子的意思，然后一步步地修改错误。在修改时选择对句子改动最小的方式来纠正上述语法错误。
输出控制 Prompt	我会以‘<句子编号>错误句子’ 的形式给出需要修改的句子，你需要以‘<句子编号>修改后的句子’ 的形式返回修改后的结果，结果仍然是中文。
一致性 Prompt	每行都是一句独立句子，不需要和下一行合并考虑，且无需输出注解，给出的行数和返回的行数必须一致。

Table 7: Prompt 示例

4.2 大模型纠错性能分析

本队伍首先分析了使用上述 Prompt 得到的大模型纠错结果，在 YACLC 开发集上简单测试对比了一下 GPT4 和 BART 模型的性能，结果如表，可以看到直接将 GPT4 用于文本纠错，其在开发集上的得分远低于常规纠错模型。但值得注意的是，这里的 $F_{0.5}$ 分值并不能准确说明 GPT4 的纠错能力。结合数据观察，能看到 GPT4 修改出了一些 BART 模型纠正不了的复杂错误，同时 BART 模型找出的语法错误 GPT4 则基本都能找出，如表 9中所示，“我认为空气污染是跟我们的生活密切的问题”为病句，BART模型并未找出这个语病，GPT则做出了正确修改。GPT4 评分低的原因在于做了很多的过度润色和改写，仍然以表 9中句子为例，GPT将“所以一定”改写为了“因此必须”，这属于不必要的过度改写。由于目前文本纠错领域标注数据集时一般只考虑语病，因此这些过度润色与改写在纠错评测中统一归于误纠。我们在mucgec数据集上的分析显示，GPT4 模型的结果中，FP (对正确token进行误纠) 的数量为常规纠错模型的三倍。

模型	最小改动维度		
	P	**R**	**$F_{0.5}$**
GPT4	47.51	51.00	48.17
Seq2Edit	59.52	35.59	52.46
Seq2Seq	67.87	39.28	59.24

Table 8: GPT4 和中小纠错模型在 YACLC 开发集上的得分

如果不考虑过度润色和改写，大模型对语法错误的检测和纠正精确率其实很高，尤其是一些复杂错误的检测和纠正已经很大程度超过了目前文本纠错的 SOTA 模型。

SRC	我认为空气污染是与我们的生活密切的问题，所以一定要最优先解决，优其是像北京那样的大城市。
BART	我认为空气污染是与我们的生活密切的问题，所以一定要~~最~~优先解决，尤其是像北京那样的大城市。
GPT	我认为空气污染是与我们的生活密切相关的问题，因此必须~~最~~优先解决，特别是像北京这样的大城市。
REF	我认为空气污染是与我们的生活密切相关的问题，所以一定要最优先解决，尤其是像北京这样的大城市。

Table 9: LLM和BART纠错模型纠错例子，蓝色和红色分布表示正确、错误修改

4.3 大模型参与集成

考虑到上述大模型特性，要将大模型用于辅助文本纠错任务，首先要解决过纠问题，本队伍尝试了一些 Prompt 去限制大模型的过纠现象，但效果都不是很好，根本原因在于难以通过 Prompt 让大模型准确理解纠错是否过度。因此我们转换思路，通过将大模型和纠错模型集成，调整大模型参与集成的权重，以此来避开大模型的过纠问题，同时利用其对语法错误的高检测率和高精确率辅助纠错。通俗来讲，我们将大模型作为判断纠错模型预测的编辑是否正确的权威专家，但限制大模型无法亲自提供纠错编辑。具体而言，在多模型投票集成中，我们将大模型的投票权重设为 3，纠错模型的投票权重设为 1，最终采纳编辑的投票阈值设为 4，即只要有任意纠错模型预测结果和大模型相同，就采纳；而纠错模型通常不会有过度改写的编辑，即使有，由于过度改写的多样性，编辑恰好和大模型重复的概率也很低，因此这种集成方式能很好地筛去大模型的过度改写编辑，同时保留那些真正有语法错误的纠正。

实验显示，这样的集成方式能为纠错系统带来 1.3% 的提升。值得一提的是，随着集成模型数量的增加，以及纠错系统性能的提高，大模型的权重需要相应调整。以本队伍实际测试的结果而言，如果纠错系统的精确率较高（如已超过 80%），且集成模型数量较多，就需要降低大模型的投票权重，只将其作为一个权重较高的子模型参与集成，换句话说，就是降低大模型的权威性。

本队伍还尝试了多 GPT 分层集成的方式，思路是既然只将大模型作为判断纠正是否正确的专家，那么先将大模型的预测结果以一个较低的投票阈值集成，增加识别的语法错误，得到召回率较高的集成结果，然后再作为判断纠错编辑是否正确的工具与常规纠错模型集成，这样能更好发挥其权威专家的作用。实验显示这样的分层集成方式对结果的提升有限，我们分析后发现多 GPT 集成对召回率的提升不明显，原因是单个大模型已能找出绝大部分语法错误，剩余的复杂错误很难通过多次使用大模型预测的方式补充找出。

模型	最小改动维度		
	P	**R**	**$F_{0.5}$**
GPT3.5 (中)+3Seq2Seq+3Seq2Edit	80.19	50.09	71.59
GPT3.5 (英)+3Seq2Seq+3Seq2Edit	79.69	51.62	71.87
GPT4 (英)+3Seq2Seq+3Seq2Edit	80.62	52.68	72.89

Table 10: 不同 Prompt 和 GPT 版本的性能差异

5 实验

5.1 训练设置

Seq2Seq 模型结构如 2.1 节所描述，使用 pytorch (Paszke et al., 2019) 库和 fairseq (Ott et al., 2019) 框架搭建，模型参数方面，学习率为5x10-4，batch size为8096 tokens，最大epoch设置为20，核心代码和详细参数设置参考 SynGEC (Zhang et al., 2022c) 开源模型，。Seq2Edit 模型结构如 2.2 节所示，首先冻结encoder，学习率设为1x10-3，训练2个epoch，然后训练全部模型参数，学习率设为1x10-5，最大epoch设为10，核心代码和详细参数设置参考 MuCGEC (Zhang et al., 2022b) 开源模型。

封闭任务的训练具体流程为: (1) 使用如 3.1 节所述的规则加噪方式，以 Lang-8 数据集目标端的正确句子为种子语料，构建 10M 伪数据预训练。(2) 使用主办方提供的 Lang-8 数据集进行微调。(2) 使用 YACLC 开发集进行精调。

开放任务的训练具体流程为: (1) Seq2Edit 使用如 3.1 节所述的规则加噪方式，以悟道语料库为种子语料，构建 10M 伪数据预训练。 Seq2Seq 使用 NaSGEC (Zhang et al., 2023) 发布的基于 100M 伪数据预训练的 BART 模型。(2) 使用 Lang-8、HSK、CGED、MuCGEC-Dev、NLPCC-2018-Test 混合数据进行微调 。(3) 使用 HSK、CGED、MuCGEC-Dev、NLPCC-2018-Test 混合数据进行进一步微调。(4) 使用 YACLC 开发集进行精调。

5.2 测试集结果

本次比赛中，我们 Seq2Seq 的单模型最高得分为 55.63，Seq2Edit 的单模型最高得分为 52.27，我们尝试了多种集成方案，同时在开放任务测试了一些大模型参与集成的方法，测试集的详细结果可参考表 (11)。可以看到，开放任务使用其他开源数据在 Seq2Edit 模型上的提升不大，在使用 YACLC 开发集精调前，混合数据能给模型带来 5% 左右的提高，但在精调后，模型最终的性能与封闭任务相差不大，我们推测是训练阶段太多，Seq2Edit 模型比较容易遗忘前面的知识，因此开放任务的集成我们最终使用了封闭任务的 Seq2Edit 模型。在 Seq2Seq 模型上，混合数据的效果较好，能带来 1.6% 的提升。

在投票集成中，我们将投票阈值设为集成模型的一半，这样能最大程度发挥两种模型的互补效果，同时保证精确率召回率的平衡。值得注意的是，开放任务中由于模型集成数量的增加，大模型参与集成的权重过高会影响精确率，因此我们降低了大模型的权重。

封闭任务	平均值	最小改动维度			流利提升维度		
	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$
Seq2Edit	51.90	70.49	51.49	65.65	44.65	24.14	38.16
Seq2Seq	54.08	74.08	54.41	69.08	46.37	24.02	39.09
3Seq2Seq+3Seq2Edit	56.20	76.21	55.75	71.00	48.52	26.08	41.40
6Seq2Seq+6Seq2Edit	59.75	83.51	54.12	75.33	54.86	24.81	44.16
12Seq2Seq+12Seq2Edit	60.59	85.44	53.44	76.3	56.53	24.6	44.88
开放任务	平均值	最小改动维度			流利提升维度		
	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$	**P**	**R**	$\mathbf{F_{0.5}}$
Seq2Edit	52.27	71.64	51.45	66.43	45.36	23.24	38.1
Seq2Seq	55.63	73.86	61.64	71.05	44.59	28.88	40.22
3Seq2Seq+3Seq2Edit	56.81	75.51	59.77	71.73	47.22	28.89	41.9
6Seq2Seq+6Seq2Edit	58.38	79.17	58.81	74.04	49.31	27.85	42.72
12Seq2Seq+12Seq2Edit	60.55	88.5	50.08	76.73	58.71	22.45	44.37
GPT4+12Seq2Seq+12Seq2Edit	61.75	87.25	54.95	78.07	55.87	26.01	45.44

Table 11: 测试集提交结果

6 结语

在本次 CCL2023-CLTC 评测任务中，我们使用了 Seq2Seq 和 Seq2Edit 两种模型，采用了伪数据——Lang-8 数据——YACLC 开发集数据的三阶段训练方案，尝试了数据增强、数据清洗、数据后处理以及模型集成技术，同时在大模型辅助纠错方面做了一些探索。实验结果表明，这些策略均可以使模型性能得到有效的提升，最终我们的纠错系统在赛道一封闭、开放任务测试集上的总得分为 60.59、61.75，均位列第一。

但是本次的系统依旧有很多不足，例如投票集成时的权重只是模型级别，未细分到错误类型级别，未来可以尝试统计模型在不同错误类型上的性能，给予不同的投票权重。此外在大模型辅助纠错方面，我们只探索了使用大模型直接预测纠错结果并参与集成的方式，并未尝试大模型知识萃取，生成伪数据的技术。

致谢

衷心感谢章岳、李嘉诚，两位在之前评测中积累了很多经验、代码和模型，是我们参加本次评测的重要基础。此工作受国家自然科学基金（62176173）资助，同时也受到江苏高校优势学科建设工程资助项目，阿里巴巴AIR计划项目支持。

参考文献

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS 32*, pages 8024–8035.

Gaoqi Rao, Qi Gong, Baolin Zhang, and Endong Xun. 2018. Overview of NLPTEA-2018 share task Chinese grammatical error diagnosis. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–51.

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.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Hang Yan, 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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NeurIPS*, 30.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*.

Yingying Wang, Cunliang Kong, Liner Yang, Yijun Wang, Xiaorong Lu, Renfen Hu, Shan He, Zhenghao Liu, Yun Chen, Erhong Yang, et al. 2021. Yaclc: A Chinese learner corpus with multidimensional annotation. *arXiv preprint arXiv:2112.15043*.

Yue Zhang, Zuyi Bao, Bo Zhang, Chen Li, Jiacheng Li, and Zhenghua Li. 2021. Technical report of suda-alibaba team on ctc-2021. Technical report.

Yue Zhang, Haochen Jiang, Zuyi Bao, Bo Zhang, Chen Li, and Zhenghua Li. 2022a. Mining error templates for grammatical error correction. *arXiv preprint arXiv:2206.11569*.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022b. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of NAACL-HLT 2022*, pages 3118–3130.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022c. Syngec: Syntax-enhanced grammatical error correction with a tailored gec-oriented parser. In *Proceedings of EMNLP*, pages 2518–2531.

Yue Zhang, Bo Zhang, Haochen Jiang, Zhenghua Li, Chen Li, Fei Huang, and Min Zhang. 2023. NaSGEC: a multi-domain Chinese grammatical error correction dataset from native speaker texts. In *Findings of ACL*.

Baolin Zhang. 2009. Features and functions of the hsk dynamic composition corpus. *International Chinese Language Education*, 4:71–79.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Proceedings of NLPCC: 7th CCF International Conference*, pages 439–445. Springer.