CCL2022-CLTC赛道二：Top1参赛系统评测报告

李云良
北京世纪好未来
liyunliang@tal.com

王智浩
北京世纪好未来
wangzhihao3@tal.com

胡飞
北京世纪好未来
hufei6@tal.com

摘要

本文描述了我们在第21届中国计算语言学大会(CCL-2022)的“汉语学习者文本纠错评测(王莹莹et al., 2022)”任务赛道二中评测排名Top1的参赛系统。在数据方面，我们主要使用了官方提供的lang8和开源的历年CGED(Chinese Grammatical Error Diagnosis)数据集。在模型方面，我们主要采用了GECToR、Sequence-to-Action模型、指针生成网络(Pointer-Generator-Network)、基于Bert+CRF的序列标注模型、基于Bert+Bi-LSTM+CRF的序列标注模型以及拼写纠错模型。此外，在模型融合方面，我们主要基于编辑动作进行投票，来选取综合预测的结果，并且加入了基于语言模型计算困惑度(PPL)的方法以及基于badcase的trick。最终我们的最佳融合结果总分为48.49%，取得了该赛道的第一名。最终我们的最佳融合结果总分为48.49%，取得了该赛道评测排名第一名的成绩。

关键词： 指针生成网络 ； 序列标注模型 ； 困惑度

CCL2022-CLTC Track2: Top1 Competition System Evaluation Report

Yunliang Li
TAL
liyunliang@tal.com

Zhihao Wang
TAL
wangzhihao3@tal.com

Fei Hu
TAL
hufei6@tal.com

Abstract

In this paper, we describe our competition system for obtaining the Top1 in the task track 2 of "Chinese learners' text error correction evaluation(Wang et al., 2022)" at the 21st China conference on computational linguistics (CCL-2022). In terms of data, we mainly used the lang8 and CGED(Chinese Grammatical Error Diagnosis) datasets provided by the official. In terms of model, We mainly used GECToR, sequence-to-action model, Pointer-Generator-Network, Sequence tag model based on Bert+CRF, Sequence tag model based on Bert+BI-LSTM+CRF and spelling correction model. In addition, at the model fusion level, we mainly select the results of model prediction based on the voting of editing actions, and add the method of computing perplexity(PPL) based on language model and tricks based on badcases. In the end, our total score of the best fusion result was 48.49%, and we won the first place in this track.

Keywords: Pointer-Generator-Network , Sequence tag model , sequence-to-action , perplexity , Bert+BI-LSTM+CRF , Bert+CRF

1 引言

文本纠错(Text Correction)任务是自然语言处理(NLP)中的经典任务之一，而且是基础任务。中文语法错误检测(Chinese Grammatical Error Diagnosis)在文本纠错领域是一个价值较高且难度较大的任务。现实中的文本数据普遍存在语病问题，这不但给文本的受众造成阅读理解的困扰，在一些人工智能任务将这类数据用于下游任务(例如对话、文本分类、实体抽取等)时，也会影响模型的性能。因此语病诊断可以作为一个上游任务，提高下游任务的数据质量，拉高模型的性能上限。在教育领域，学生需要学习怎样正确的使用语法，所以学生在学习过程中产生的文本大概率会存在一定的语病问题。但是人工进行文本校对，工作复杂效率极低。中文语法错误检测能高效率地指导学生提升语法的使用技能。综上来说，该任务尤为重要。

在使用算法解决中文语法错误检测的任务时，将中文语法错误进行抽象设计，共抽象出4类语法错误。语法错误的类型分为赘余(Redundant Words，R)、遗漏(Missing Words，M)、误用(Word Selection，S) 、错序(Word Ordering Errors，W) 四类。本次比赛目的是检测出中文文本中每一处语法错误的位置、类型。评测任务要求参加评测的系统输入句子(群)，其中包含有零个到多个错误。参赛系统应判断该输入是否包含错误，并识别错误类型，同时标记出其在句子中的位置和范围，对缺失(M)和误用(S)给出修正答案。

参赛系统融合了多个模型，来解决中文语法错误检。模型中一类是语病位置检测模型，使用MacBert-large+CRF和MacBert-large+BiLSTM+CRF两种方法，仅提取句子中错误的位置以及错误类型。一类是基于序列标注的语病纠错模型GECToR获取句子的纠正序列。另一类是基于指针生成网络，使用不同的训练策略，得到的2个语病纠错模型：最后还有一类单独针对错别字的模型。另外，我们复现了腾讯在AAAI 2022发表的语法纠错论文中的Sequence-to-Action模型，虽然在最高分的融合方案中没有加入该模型的结果，但在比赛过程中该模型提供了较大的贡献。

在参与比赛的过程中，我们主要有以下几点发现: (1)针对不同类型的模型，数据集的训练策略不同会产生不同的训练效果，例如lang8和历年数据集合并训练或者分步训练在同一类模型的表现有较大差距; (2)在数据集的处理层面，由于这一届评测任务和往届不同，错误是基于字符级别，且都是简体字，因此事先将数据集统一处理为字符级别和简体字会更有效果; (3)在基于文本生成的纠错模型中，提高性能指标的关键点在于如何降低过纠正的问题，即是尽可能将对的字段拷贝过去，将错的字段重新生成; (4)在模型融合过程中，模型之间的差别越大、种类越多，融合效果会更好，且结合某个模型的优势来筛选结果也有一定的提升。

我们提交的参赛系统包含语病检测、语病纠错以及拼写纠错模型，不同的模型具有各自的优点，对于某个类型的数据修改的效果更好，在将这些模型各自的结果融合后，该参赛系统相当于整合了各个模型的优点，具有更好的泛化性能。但是，由于本身包含多个模型，并且其中有基于文本生成的语病纠错模型，从而导致该参赛系统的推理耗时较高，占用资源较高，不适用于业务工程化上线。

我们将参赛系统开源在github上https://github.com/AI-confused/CCL2022_CGED-8_Top1_project

2 背景

汉语学习者文本纠错任务(Chinese Learner Text Correction，CLTC)旨在自动检测并修改汉语学习者文本中错误。

2.1 任务设置

该任务中模型需要根据输入的纯文本内容，输出语病检测结果，其中对于“S”和“M”类型的错误要给出纠正结果(一个或多个)。

输入 (sid=00038800481) 我根本不能了解这妇女辞职回家的现象。在这个时代，为什么放弃自己的工作，就回家当家庭主妇?

输出 00038800481, 6, 6, S, 理; 00038800481, 8, 8, R (“了解”应为“理解”，删去“这”)

Table 1: lang8和CGED历年数据的单句错误个数分布

model		0	1	2	3	4	5	大于5
lang8	数量	113055	381623	297212	165275	79474	35709	28033
	占比	10.27%	34.68%	27%	15.02%	7.22%	3.25%	2.55%
CGED	数量	6021	25394	9841	5815	3508	1886	2292
	占比	11%	46.38%	17.97%	10.62%	6.41%	3.44%	4.19%

Table 2: lang8和CGED历年数据的各错误类型分布

model		M	S	R	W
lang8	数量	720020	944435	444444	51411
	占比	33.33%	43.72%	20.57%	2.38%
CGED	数量	34479	36676	24292	7350
	占比	33.54%	35.68%	23.63%	7.15%

评价指标 根据输入输出设定了6个方面的性能评价，分别是：(1)假阳性(False Positive)：正确句子被判包含错误的比例；(2)侦测层(Detective-level)：对句子是否包含错误做二分判断；(3)识别层(Identification-level)：给出错误点的错误类型；(4)定位层(Position-level)：对错误点的位置和覆盖范围进行判断，以字符偏移量计；(5)修正层(Correction-level)：提交针对字符串误用(S)和缺失(M)两种错误类型的修正词语。修正词语可以是一个词，也可以是一个词组；(6)综合打分(Comprehensive Score)：本年CGED-8引入前5项指标的加权平均分数作为综合打分，最终以综合分数排名决定奖项。

2.2 评测数据集情况

lang8数据集共包含1100381个句子对，但错误句子去重后共654072句，修改后的正确句子为1076213份，说明同一个错误句子，可以有多种修改方案，这也是此任务的一个难点。经过新的错误解析代码处理后，lang8数据和CGED历年可用数据集，单句错误个数分布如表1所示，可见lang8数据中出现1个及1下错误的句子占比44.95%，单句出现多个错误的情况较多，这对模型的训练是一个考验。CGED数据中对于1个错误的情况也很多。lang8数据和CGED历年数据各错误类型分布如表2所示，可见在两批数据上，S类型占比最大，W占比较小。

2.3 相关工作

语言在人类社会发展过程中，起到了非常大的作用，语言的学习也就至关重要。对于儿童或者学习非母语的人而言，在语法方面经常会出错。所以在教育领域，语病诊断是一个亟待解决的课题。

在语病诊断的相关研究中，最早出现的方法是基于规则模版的(Foster and Vogel, 2004)，即根据特定的语料以及语病的特征，设计构造算法规则，来识别出文本中的语病。这种方法存在一定的弊端：设计构造规则需要耗费大量的人力，且规则的泛化性较差。

然后，随着机器学习的兴起，研究人员开始将机器学习方法应用到语病诊断任务(Rei and Yannakoudakis,)。该类方法基于人为构造的特征，运用机器学习模型对文本进行语病诊断，效果优于基于规则的方法。然而，人为构造特征仍然需要耗费很多的人力，且整体指标效果偏低。

近些年兴起的深度学习，促进了语病诊断任务的进步，以数据驱动的深度学习方法在效果上要优于基于机器学习人为构造特征的方法，已有的方法主要有：(1)采用分类器来诊断文本的语病(Han et al., 2004)，这种方法比较局限，因为单个样本中可能包含多种语病类别，且较难识别出语病的位置信息；(2)采用序列标注方法对文本进行字符级别的诊断(Zheng et al., 2016)，这种方法对文本的字符语义信息进行编码，抽取出语病的类别与位置，然而样本中同一个字符位置可能有多种语病标签，这又会引入实体重叠的问题，增加了训练的难度，并且目前此方法的效果并不理想；(3)采用文本生成的方法对文本进行语病纠错(Yuan and Briscoe, 2016)，生成语法正确的文本，再结合源文本获取语病诊断的语病类别和位置的输出，该方法需要大量的“源

文本—目标文本”的句对样本来训练文本生成模型，且文本生成的输出可控性较低，可能会生成偏离源文本较大的结果，导致较难获取真实语病信息。

2.4 贡献

总的来说，我们的贡献有以下几点:

- 结合文本摘要的思路，将指针生成网络用于语病纠错模型对病句进行纠错，既最大程度地保留了正确的字段，又对错误的字段进行了修改，且单模型指标较高。
- 针对基于指针生成网络的语病纠错模型生成[UNK]字符的问题，结合指针网络的注意力机制，将[UNK]替换为输入序列中注意力最高的字符，从而提高纠错模型的性能。
- 在错别字模型中，设计了持续增强策略，可以在每次训练迭代中，生成新的训练数据。
- 在模型融合方案上，基于编辑动作的投票来筛选票数悬殊的结果以及计算错误点修改前后的困惑度降低值来筛选票数均衡的结果。

3 参赛系统

在这个章节我们会详细阐述我们在参赛系统中用到的方案思路，主要分为一下几个部分：(1)数据预处理; (2)语病检测模块; (3)语病纠错模块; (4)拼写纠错模块; (5)模型融合模块。

3.1 数据预处理

lang8 (1)由于lang8数据本身存在噪音，因此先后使用繁体转简体、清洗特殊字符串、清洗重复标点、清洗emoji等操作对原始数据进行清洗；(2)剔除部分无效的样本，包含：src和correct长度差异过大；文本长度过短；中文字符占比低于20%；(3)存在部分样本的src和correct末尾标点缺失，这类样本的“M”标签不好处理，因此统一补上标点，剔除该错误点；(4)使用pair2edit文件，将lang8文件生成char级别的错误编辑，最后转换成jsonl格式的数据集;

CGED历年数据 (1)CGED历年数据集种类繁杂，有简体和繁体，因此需要统一将繁体转成简体，但有个比较明显的问题：“着”会被误转成“著”，这个占比不低，因此需要手动将误转的“著”换成“着”；(2)2022届强调的错误点基于字符级别，而历年数据集的错误点大多基于词语级别，因此在训练任务之前使用官方提供的pair2edit文件进行转换，统一将字符级别的错误标签替换掉词语级别的错误标签。其中16年、17年、18年的测试集没有提供correct字段，无法使用pair2edit方法，因此沿用原本的错误标签即可；(3)删除部分标注问题较大的数据，清除文本长度过短的数据；(4)将14年-18年的训练集和测试集作为参赛系统中CGED领域的训练集，将20-21年的测试集作为参赛系统的验证集;

3.2 语病检测模块

该任务的目的是识别出输入文本中的错误位置及其错误类型，可以采用序列标注方案来解决该问题，这里我们采用Bert+CRF与Bert+Bi-LSTM+CRF的方案。

Bert+CRF 模型结构如图 1所示，将输入序列经过BERT(Devlin et al., 2019)编码获取每个字符的表征信息，在经过一层分类器后获取每个字符在语病标签空间的分布，将其作为CRF层的发射矩阵，最后得到输入序列的语病标签输出序列，这里CRF层能更好地考虑语病标签之间的关系，提高模型的性能。

标签方面我们采用了BIESO格式的标签，共17种标签，采用类型(RMSW)-位置(BIES)的格式，用O表示该字符无错误，语病标签空间为’O’, ’S-B’, ’M-B’, ’W-B’, ’R-B’, ’S-I’, ’M-I’, ’W-I’, ’R-I’, ’S-E’, ’M-E’, ’W-E’, ’R-E’, ’S-S’, ’M-S’, ’W-S’, ’R-S’。

Bert+Bi-LSTM+CRF 基于上述方案，为了更好地对文本序列之间的联系进行编码，我们在Bert和CRF之间添加了一层Bi-LSTM网络。模型结构如图2所示，将输入序列经过BERT编码获取每个字符的表征信息，随后输入Bi-LSTM网络进一步获取字符之间的上下文表征，在经过一层分类器后获取每个字符在语病标签空间的分布，将其作为CRF层的发射矩阵，最后得到输入序列的语病标签输出序列。

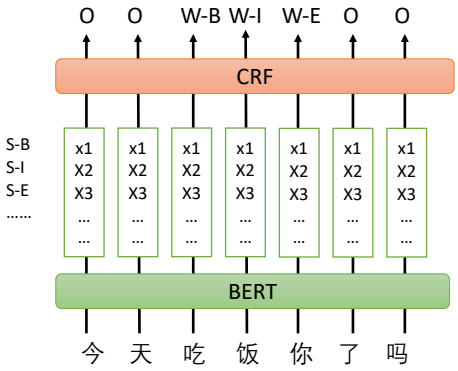

Figure 1: BERT+CRF模型结构

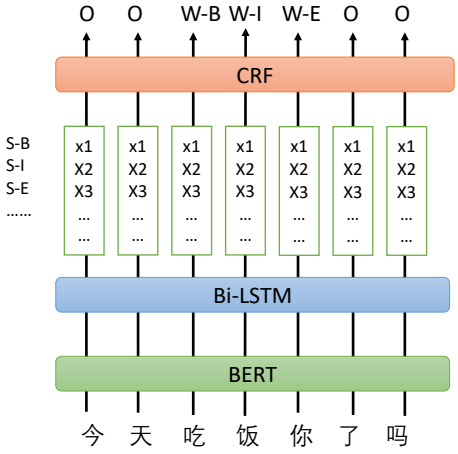

Figure 2: BERT+Bi-LSTM+CRF模型结构

标签方面我们采用了BIESO格式的标签，共13种标签，采用类型(RMSW)-位置(BIES)的格式，用O表示该字符无错误，且把一些不可能出现的标签去掉了(M类型只有Single，W类型没有Single)，语病标签空间为'O', 'S-B', 'M-B', 'W-B', 'R-B', 'S-I', 'M-I', 'W-I', 'R-I', 'S-E', 'M-E', 'W-E', 'R-E', 'S-S', 'M-S', 'W-S', 'R-S'。

3.3 语病纠错模块

语病检测模型虽然可以直接给出文本中的错误位置及错误类型，但是针对S和M类型的错误无法给出纠正结果，因此可以采用基于文本生成的语病纠错方案生成输入错误文本的纠正文本，然后使用转换函数将原文本和纠正文本输出错误位置及其错误类型，这样S和M类型的错误能给出纠正结果。在语病纠错模块中我们探索了GECToR (Omelianchuk et al., 2020)、Sequence-to-Action(Li et al., 2022)、Pointer-Generator-Network(See et al., 2017)方法，其中基于文本摘要思路的Pointer-Generator-Network方法的性能超出了预期。

GECToR 该方法是一种基于序列标注的语病纠错模型，采用seq2edit思想解决语病修改问题，模型结构如图3所示，对一个源句子序列使用编码器进行编码，然后在每个字符处使用分类器预测最可能的编辑标签，对于序列中的每个字符都映射到一个编辑操作，编辑操作包括"KEEP"，"DELETE"，"APPEND"，"REPLACE"，分别代表"保存当前字符"、"删除当前字符"、"当前字符后面添加相应的字符"、"当前字符替换成相应的字符"，其中"KEEP"和"DELETE"是单标签，"APPEND"和"REPLACE"需要拼接添加或者替换的字符，比如"REPLACE_自"和"APPEND_下"。

Sequence-to-Action 该方案是腾讯在AAAI-2022发表的一篇中文纠错领域SOTA论文，该方案的思路主要是在传统的Transformer上添加了一个动作分类模块，基于每个字符的动作分类结

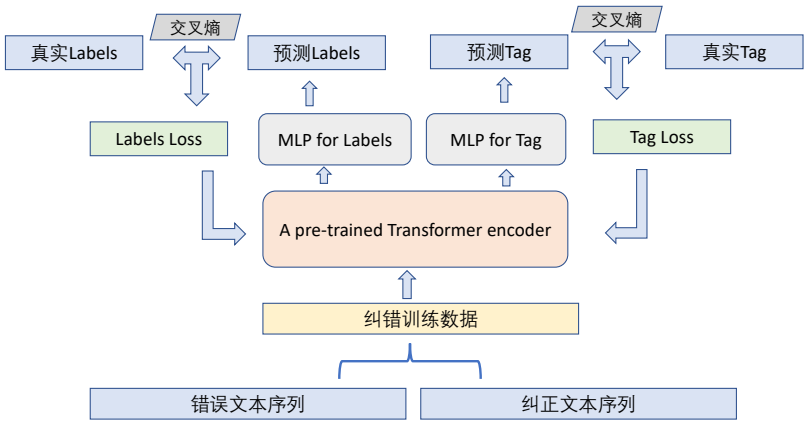

Figure 3: GECToR模型结构，一种基于序列标注的语病纠错模型

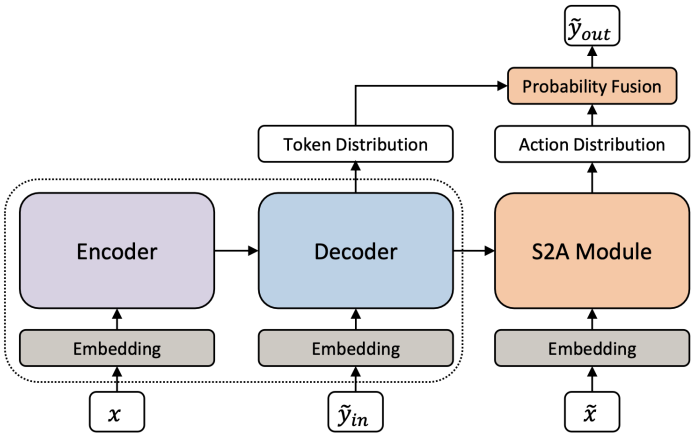

Figure 4: Sequence-to-Action模型结构(Li et al., 2022)

果来指导文本生成结果，模型结果如图4所示。该论文的出发点是，在基于文本生成的语病纠错任务中，需要解决的核心问题为如何降低过纠正情况，即尽可能保留正确的内容，只生成需要修改的内容。模型在文本生成的字符分布概率基础上融合了S2A模块的字符动作分布概率，动作空间为“SKIP”、“GEN”、“COPY”。如果一个字符在S2A模块被预测为“COPY”，说明该字符应该保持不变；如果被预测为“SKIP”，则该字符的输出应为[BLK](空字符，表示该位置为空)；如果被预测为“GEN”，则输出字符取决于文本生成的结果。这样便避免了文本生成导致的过纠正问题，提高了纠错模型的性能。

Pointer Generator Network 指针生成网络是一种生成式文本摘要方法，在生成摘要时允许生成新的词语来组成摘要，也可以直接拷贝原文中的内容构成摘要，这种任务形式与语病纠错基本保持一致，当遇到正确的字段时需要直接拷贝到结果中，当遇到错误的字段时则生成正确的字段，并且指针网络天然地适用于乱序类型的错误。该模型结构如图5所示，它混合了seq2seq和Pointer Network(Vinyals et al., 2015)，它具有seq2seq的生成能力和Pointer Network的拷贝能力，在生成纠错结果时同时考虑seq2seq的生成概率和输入序列的注意力分布概率，更好地解决纠错模型的过纠正问题，提升模型性能。

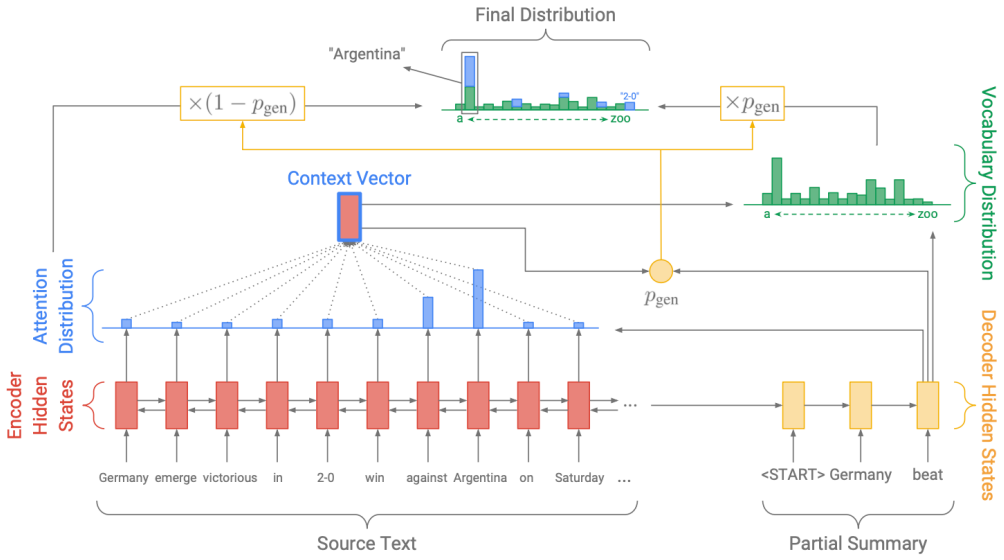

Figure 5: Pointer-Generator-Network模型结构(See et al., 2017)

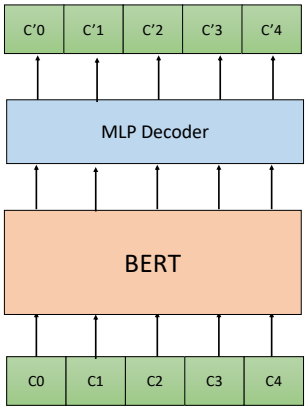

Figure 6: 拼写纠错模型结构

3.4 拼写纠错模块

该模块使用了2个错别字模型，分别是**近形音错别字模型**和**病句错别字模型**，均为track1的baseline模型，模型结构如图6所示，区别是训练策略不同。近形音错别字模型主要是针对常见错别字、形近字、音近字的单字错误情况，而病句错别字模型针对的是病句中的错别字情况，且仅针对修正字符数小于等于原字符数的情况。

3.5 模型融合模块

参与最终融合结果的模型一共有7个，分别是：1个GECToR、2个指针生成网络(Pointer-Generator-Network)、1个基于Bert+CRF的序列标注模型、1个基于Bert+BiLstm+CRF的序列标注模型以及2个拼写纠错模型，融合的思路主要是：(1)将所有模型的预测结果合并，得到一个包含所有模型结果的集合；(2)对于判定一个样本是否为correct，主要基于模型对该样本的投票，当不小于3个模型预测该样本为correct，则认为该样本为correct；(3)多个模型融合结果中存在同一个位置有多个错误类型的情况，这种情况可以采用参与判定模型的数目以及计算修改前后的困惑度降低值来进行筛选；(4)多个模型融合结果中存在位置有重叠的情况，这时可以采用参与判定模型的数目来筛选；(5)最后对于S和M的纠正结果，使用一些基于验证

Table 3: 语病检测模型的实验结果

model	phase	dataset	FPR	detect-f1	identification-f1	position-f1
BERT+CRF	lang8预训练	20+21测试集	0.2144	0.7989	0.5301	0.3004
		22测试集	0.2212	0.7836	0.5135	0.3058
	CGED微调	20+21测试集	0.1642	0.7977	0.5568	0.3469
		22测试集	0.1740	0.7723	0.5093	0.3339
BERT+Bi-LSTM+CRF	lang8预训练	20+21测试集	0.1858	0.8007	0.5318	0.3014
		22测试集	0.2183	0.7780	0.4978	0.3013
	CGED微调	20+21测试集	0.1799	0.8083	0.5635	0.3514
		22测试集	0.2094	0.7753	0.5089	0.3298

集badcase的trick，例如删除长度超过3的结果、删除字符重复率过高的结果等。

4 实验部分

在这个章节，我们会详细阐述在参赛过程中的实验部分，主要分为语病检测、语病纠错、拼写纠错、badcase分析和模型融合。

4.1 语病检测模块

数据集 在语病检测模块中，我们主要采用了数据集分步训练的方法，即先用lang8预训练、再用CGED历年数据微调模型。

超参数设置 模型中的BERT采用的是Chinese-MacBERT-large(Cui et al., 2020)，该预训练语言模型在中文任务上能有更好的效果。

BERT+CRF的超参数设置: lang8预训练阶段，我们设置的学习率为1e-5，输入序列最大长度为256，训练批度为256，dropout为0.1，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮; CGED历年数据集微调阶段，我们设置的学习率为5e-6，输入序列最大长度为256，训练批度为256，dropout为0.3，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮。

BERT+Bi-LSTM+CRF的超参数设置: lang8预训练阶段，我们设置的学习率为1e-5，输入序列最大长度为256，训练批度为256，dropout为0.1，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮；CGED历年数据集微调阶段，我们设置的学习率为5e-6，输入序列最大长度为256，训练批度为256，dropout为0.3，全局随机种子为99，采用了linear的学习率调整器，且预热比例为0.1，一共训练10轮。

实验结果 语病检测模型在验证集和测试集的检测指标如表3所示，可以看出经过lang8和CGED分步训练后，模型的整体指标增高，且验证集和测试集的指标差距并不大。

4.2 语病纠错模块

数据集 在语病纠错模块中，GECToR模型使用的是先用lang8预训练、再用CGED历年数据微调模型；Sequence-to-Action模型使用的是lang8+CGED历年数据的单步训练方法，该模型结果未参与最终提交结果的融合；Pointer Generator Network分别采用了2种训练策略得到2个模型，一个使用的是lang8+CGED历年数据的单步训练方法，另一个基于前者的最佳模型在CGED历年数据上进行了二阶段微调。

超参数设置 Pointer Generator Network和Sequence-to-Action模型的Seq2Seq模块采用的均为Chinese-Bart-Large(Shao et al., 2021)，GECToR的编码器模块采用的是Chinese-Struct-Bert-Large(Wang et al., 2020)。

Sequence-to-Action的超参数设置: 我们设置的学习率为5e-5，输入序列最大长度为256，训练批度为256，dropout为0.1，损失权重alpha为0.001，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮。

Pointer Generator Network单步训练的超参数设置: 我们设置的学习率为5e-5，输入序列最大长度为256，训练批度为256，dropout为0.1，beam search的k为3，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮。

Table 4: 语病纠错模型的实验结果

model	phase	dataset	FPR	detect-f1	identification-f1	position-f1	correction-f1	com
Sequence-to-Action	lang8+CGED单步	20+21测试集	0.2625	0.8279	0.4909	0.25	/	/
		22测试集	0.2552	0.8273	0.5056	0.2745	0.1673	0.3799
Pointer-Generator-Net	lang8+CGED单步	20+21测试集	0.077	0.6837	0.443	0.2605	/	/
		22测试集	0.0796	0.6995	0.4602	0.2997	0.2070	0.3967
Pointer-Generator-Net	CGED微调	20+21测试集	0.1976	0.8136	0.5673	0.3517	/	/
		22测试集	0.1431	0.80	0.5456	0.3553	0.2505	0.4521
GECToR	lang8预训练	22测试集	0.2684	0.7460	0.4691	0.2839	0.1858	0.3541
	CGED历年微调	22测试集	0.3215	0.8054	0.5415	0.3363	0.2268	0.3971

$$m = \begin{cases} 0 & , n < 5 \\ 1 + \left\lfloor \dfrac{m-5}{10} \right\rfloor & , n \geq 5 \end{cases}$$

Figure 7: 错别字生成策略公式

Pointer Generator Network二阶段微调的超参数设置：我们设置的学习率为5e-6，输入序列最大长度为256，训练批度为256，dropout为0.1，beam search的k为3，全局随机种子为99，采用了cosine的学习率调整器，且预热比例为0.0，一共训练10轮。

GECToR的超参数设置：第一步lang8预训练中，冷启动阶段只有MLP for Labels层和MLP for Tag层参与训练，我们设置的学习率是1e-3，训练批度是128，训练轮数是2，非冷启动阶段所有模型参数参与训练，我们设置的学习率是1e-5，训练批度是48，训练轮数是20，patience是3；第二步CGED历年数据微调中我们设置的学习率是1e-5，训练批度是48，训练轮数是20，patience是3，全局随机种子设置都是1。

实验结果 语病纠错模型在验证集和测试集的整体指标如表4所示，可以看出在使用CGED历年数据进一步微调后指针生成网络的单模总分可以高达0.4521。

4.3 拼写纠错模块

近形音错别字模型 我们从公开数据集中获取一个错别字库，库中每个字有三组不同类型的易错字对象，常见错别字、形近字、音近字。其中常见错别字是从任务一公布的训练集和验证集，还有开源资源[0] 收集而得；形近字从开源资源[1] 获取；音近字取lang8数据、任务一训练集和验证集所有出现汉字的top4000，使用拼音工具收集而得。

我们用到了一种训练增强策略，选取lang8数据中1000条数据作为验证集，其余数据作为训练集。在训练过程中，每个epoch会将所有数据的错误重新生成一次。生成策略为：如果每个句子中含有n个中文字，则生成m个错别字，公式如图7所示。

然后在中文字符索引随机选择m个字符，对于每个选中的字符生成错别字。首选依概率选取错别字类型，其中生成常见错别字、形近字错别字、音近字错别字概率分别为为50%，15%，35%，选定类别后，如果选定字存在选定的错别字类型，则在错别字列别中随机选择一个字进行替换。

病句错别字模型 本模型采用lang8数据来构造数据进行模型训练，数据构造过程为：

(1)每条目标样本选取一个病句样本：我们发现在lang8数据中，存在一个目标样本，对应多种病句样本的情况。采用编辑距离等策略，在1对多的情况下，选择一种错误情况，匹配成对；

(2)提取R、S类型数据：对获取的字符串对，使用任务二提供的格式转换代码，转换成S、M、R、W错误类型的格式；

(3)格式转换：在此模型训练时，我们仅加入了R、S两种类型，数据处理方式为：[1]将所有R对应的字符，都替换成“[unused1]”；[2]针对S类型，修正结果和原串错误位置长度相等，则

[0]https://github.com/onebula/sighan_raw
[1]https://github.com/shibing624/pycorrector

Table 5: 最终融合的实验结果

com	FPR	detect-f1	identification-f1	position-f1	correction-f1
48.49%	14.9%	79.94%	60.78%	39.85%	28.31%

等长替换，如果修正结果短于原串错误位置长度，则将缺失位置用“[unused1]”替换，如果修正结果长于原串错误位置长度，则将答案长的部分直接截取与原串位置等长。

将原串按上述处理，生成的字符串作为目标字符串，进行训练。

4.4 badcase分析

在对不同模型在验证集上的badcase分析及在“W、S、R、M”4种错误类型各自的指标比对中，我们发现指针生成网络在W类型的指标较高，且W类型的badcase较少，验证了我们最初选择该类模型的初衷，且该类模型的FPR偏低；错别字模型在字数对等的S类型纠正效果更好；Bert+CRF和Bert+Bi-LSTM+CRF这两个语病检测模型则在S类型的指标较高，对S类型的检测效果更好；GECToR模型则各类错误指标较均衡，FPR偏高，存在一定的过纠正问题。M类型的错误可能由于产生了句法的断点，导致这几种模型在该类型的指标均不高，badcase中存在较多M类型未被识别纠正，这也是后续需要探索的一个点。

4.5 模型融合

参考3.5的融合策略，将这些模型结果进行融合，得到的最高分实验结果如表5所示。

5 总结

在本次评测任务中，我们使用了序列标注模型来识别错误地址和错误类型。针对错别字提出了一个新的数据增强方法，同时首次将指针生成网络引入语法纠错任务，同时还训练了GECToR模型。融合多个模型后，我们的指标达到48.49%的成绩，位列评测排名第一。

虽然成绩达到第一，但是离落地应用还有一定距离。首先，整体指标还有待进一步提高。其次，在整个评测过程中，我们缺少对更细致的分析，以及各个模型的优劣评价，后续可以详细分析各模型的优缺点，融合各模型的优点，进行模型结构上更大的创新，来适配语法纠错任务。

在中文语病诊断领域后续的研究中，可以从如何更好地解决过度纠正问题上继续往下探索，以及从语法树层面给出辅助内容，目前该领域的SOTA指标依然未达到业务上线的基准，还需要大家一起努力探索。

6 致谢

最后，感谢北京世纪好未来教育科技有限公司的支持，感谢美校的支持，感谢智学云的支持，感谢部门领导和同事的鼓励与帮助，谢谢！

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.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jennifer Foster and Carl Vogel. 2004. Parsing ill-formed text using an error grammar. *Artificial Intelligence Review*, 21(3):269–291.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2004. Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus. In *LREC*.

Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation.

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, Seattle, WA, USA → Online, July. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. Compositional sequence labeling models for error detection in learner writing.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

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*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Jiangnan Xia, Liwei Peng, and Luo Si. 2020. Structbert: Incorporating language structures into pre-training for deep language understanding.

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.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Bo Zheng, Wanxiang Che, Jiang Guo, and Ting Liu. 2016. Chinese grammatical error diagnosis with long short-term memory networks. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 49–56.

王莹莹, 孔存良, 刘鑫, 方雪至, 章岳, 梁念宁, 周天硕, 廖田昕, 杨麟儿, 李正华, 饶高琦, 刘正皓, 李辰, 杨尔弘, 张民, and 孙茂松. 2022. Cltc 2022：汉语学习者文本纠错技术评测及研究综述.