CCL23-Eval 任务7赛道一系统报告：基于序列到序列模型的自动化文本纠错系统

刘世萱　　刘欣璋　　黄钰瑶　　王超　　宋双永
中国电信数字智能科技分公司
{liusx14, liuxz2, huangyy121, wabgc17, songshy}@chinatelecom.cn

摘要

本文介绍了本队伍在CCL-2023汉语学习者文本纠错评测大赛赛道一中提交的参赛系统。近年来，大规模的中文预训练模型在各种任务上表现出色，而不同的预训练模型在特定任务上也各有优势。然而，由于汉语学习者文本纠错任务存在语法错误复杂和纠错语料稀缺等特点，因此采用基于序列标记的预训练文本纠错模型来解决问题是自然的选择。我们的团队采用了序列到序列的纠错模型，并采取了两阶段训练策略，设计了一套基于序列到序列文本纠错的pipeline。首先，我们对训练集数据进行了清洗处理；在第一阶段训练中，我们在训练集上使用数据增强技术；在第二阶段，我们利用验证集进行微调，并最终采用多个模型投票集成的方式完成后处理。在实际的系统测评中，我们提交的结果在封闭任务排行榜上超出baseline模型17.01分（40.59->57.6）。

关键词：　序列到序列；文本纠错；语言模型

System Report for CCL23-Eval Task 7 Track 1: Automated text error correction pipeline based on sequence-to-sequence models

Shixuan Liu　　Xinzhang Liu　　Yuyao Huang　　Chao Wang　　Shuangyong Song
China Telecom Corporation Ltd. Data&AI Technology Company
{liusx14, liuxz2, huangyy121, wabgc17, songshy}@chinatelecom.cn

Abstract

This paper presents our results in Track 1 of the CCL-2023 Chinese Learner Text Correction Assessment Contest. In recent years, large Chinese pre-trained models have performed well on various tasks, while different pre-trained models have their own advantages on specific tasks. However, due to the complexity of grammatical errors and the scarcity of error correction corpus in text correction tasks for Chinese learners, it is a natural choice to use sequence-to-sequence pre-trained language models to solve the problem. Our team adopts a sequence-to-sequence error correction model and adopts a two-stage training strategy to design a text error correction pipeline. First, we clean the training dataset; We use data augmentation techniques during the first training stage; in the second stage. Then, we use the validation dataset for fine-tuning, and finally, we post-process the results and use an ensemble method by voting. In the actual system evaluation, our results outperformed the baseline model by 17.01 scores (40.59->57.6) on the closed track leaderboard.

Keywords: Sequence-to-Sequence , Grammatical text correction , Language models

1 引言

写作是一种学习技能，对非中文母语使用者而言尤为具有挑战性。我们都会偶尔在标点符号、拼写和用词方面出现小错误。虽然我们在中文母语中也会偶尔犯一些标点符号和选词不当的错误，但非中文母语作者往往更难创造出语法正确且易于理解的文本。随着自然语言处理领域的日益发展，文本自动纠错技术的重要性也变得越来越突出。这项技术可以使系统自动检测句子中的语法错误，并进行修正，从而提高文本的质量和可读性。中文文本纠错是自然语言处理(NLP)领域的一个重要任务，然而，由于中文语法和表达方式的复杂性，以及数据集规模和质量的限制，中文文本纠错仍然面临着许多挑战。

目前，中文文本纠错数据集的数量相对较少，主要来源于lang8平台的中文数据语料库。lang8平台的语料库虽然为中文文本纠错提供了一定的基础数据，但它们的覆盖范围有限，仅标注了汉字、词语和短语中的错误。这意味着现有的模型在训练过程中往往只能学习到表层错误，而无法充分修正语法错误。

在早期的文本纠错任务中，人们主要借助基于手动编码的规则来进行纠错，这些规则应用于具有鲁棒性的解析器，并通过这些解析器对文本错误进行纠正(Leacock et al., 2009)。同一时期，研究人员开始探索数据驱动的方法，使用带有示例更正的错误文本语料库和监督机器学习模型来进行修正(Rozovskaya and Roth, 2010)。随着深度学习技术的发展，各种大规模预训练模型也被用于文本纠错任务，基于Transformer的编码解码器结构模型在错误文本到正确文本的“翻译”过程中也有着较好的效果，如BART (Lewis et al., 2020)和T5(Xue et al., 2021)。一些基于Transformer解码器结构的大型生成式模型也在类似任务中有着不错的性能，如bloom(Scao et al., 2022)。深度学习模型，尤其是序列到序列(Seq2Seq)模型，在训练过程中通常需要大量的训练数据。然而，由于中文语法结构的复杂性和多样性，以及不同领域的文本特点差异较大，获取大规模且高质量的中文文本纠错数据集仍然是一个挑战。此外，现有的模型在处理复杂的语法结构时也存在困难，例如长句、多义词等。

因此，在本次CCL-2023汉语学习者文本纠错评测赛道一封闭任务中，我们基于序列到序列的模型，设计了自动化文本纠错的pipeline，实现了数据清洗和预处理，基于语法错误分布的训练数据增强，基于序列到序列模型的一阶段训练和二阶段微调，生成文本数据的后处理和不同模型的投票集成。最终，我们的模型在最小改动和流利度提升两个任务的平均F0.5得分上，超过baseline模型17.01分（40.59->57.6）

2 相关工作

在模型层面，我们的目光主要聚焦在基于Transformer架构的深度学习语言模型上。目前主流的技术主要分为两种：（1）基于序列到编辑（Seq2Edit）的语法纠错。（2）基于序列到序列（Seq2Seq）的语法纠错。基于序列到编辑的中文语法纠错方法是一种基于神经网络的方法，它通过预先定义一些编辑动作，采用神经网络为句子的token打上编辑标签，将语法纠错任务转化成序列标签任务，从而进行语法纠错。当前较为先进的模型为GECToR(Omelianchuk et al., 2020), 作为一个序列标注模型，其解码空间涵盖了插入、删除和替换等编辑操作。GECToR在训练过程中预测这些编辑，并通过后处理将其应用于原始语句。基于序列到序列的语法纠错模型则利用神经网络学习输入和输出之间的映射关系，直接输入原始错误句子到模型，模型直接输出改正后的句子，从而实现对文本中的语法错误的自动纠正的方法。这类模型通常包括编码器(Encoder)和解码器(Decoder)两个部分。编码器负责将输入的文本序列转换为一个固定长度的向量表示，这个向量包含了输入文本的所有信息。解码器则根据编码器的输出生成一个新的、正确的文本序列。在这个过程中，解码器会根据当前的上下文和已经生成的字符来预测下一个字符，如bart(Lewis et al., 2020), t5模型(Xue et al., 2021)。随着近期超大规模语言模型的流行（如chatgpt），使用超大参数量Transformer decoder only结构的生成式模型也在各个任务上表现出不错的性能，一些论文也探讨了如何在语法纠错任务上发挥超大模型的性能。在一些研究中，作者尝试设计更加精准的prompt模板来挖掘超大模型学习到的知识，从而更好地纠正原始句子中的语法错误(Fang et al., 2023)。

在文本语法纠错任务中，数据增强技术可以用于扩充训练集，提高模型的性能。常见的数据增强技术包括噪声增强、样例生成等。噪声增强是一种常见的数据增强技术，通过将训练集正确文本中的一些token随机进行替换，插入，删除等操作，形成新的错误句子来扩充训练集。例如，我们可以将原始文本中的某些单词替换为同义词或随机生成的新单词，或者在文本中添

第二十二届中国计算语言学大会论文集，第213页-第219页，哈尔滨，中国，2023年8月3日至5日。

加一些无关紧要的信息。假设我们有一个正确句子“现在我还没有吃饭呢。”，我们可以使用语义替换技术生成错误句子“现在我还没有饭呢。”。此外，我们还可以在模型参数层面添加一些噪声，从而提高模型的鲁棒性，例如使用对抗训练的方式，在模型的embedding层添加扰动。样例生成则通过生成新的样本来扩充训练集。例如，我们可以使用模板生成器或变分自编码器等技术来生成新的样本，这些样本可能与原始数据相似但不完全相同。这种方法可以帮助模型学习到更多的上下文信息，并提高其对未知数据的适应能力。

3 任务介绍

汉语学习者文本纠错(Chinese Learner Text Correction, CLTC) 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。多维度汉语学习者文本纠错则考虑到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，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。赛道一的数据中提供针对一个句子的多个参考答案，并且从最小改动（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。其中，训练集来自NLPCC2018-GEC发布的采集自Lang8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，数据统计如表格1所示。

Table 1: 开发集和测试集数据统计

	YACLC-Minimal-Dev	**YACLC-Minimal-Test**	**YACLC-Fluency-Dev**	**YACLC-Fluency-Test**
原句数	1,839	7,296	1,839	5,515
参考句数	15,938	42,462	3,332	10,237
平均参考句数	8.67	5.82	1.81	1.86
有修改的参考句数（比例）	15,935 (99.98%)	40,334 (94.99%)	3,332 (100.00%)	8,604 (84.05%)
原句平均字符数	25.85	21.19	25.85	20.81
参考句平均字符数	27.22	23.25	27.14	21.40

评价指标使用中文语法错误纠正评估工具ChERRANT,对预测结果进行评估。通过对比预测结果编辑和参考答案标准编辑，计算预测结果的精确度、召回度和F0.5值。

4 算法

我们在本次比赛中，基于序列到序列的模型设计了自动化文本纠错的pipeline，总共包含五个模块：数据预处理，数据增强，模型训练+二次微调，生成结果后处理和模型集成，具体流程如图1所示。

4.1 数据预处理

在处理lang8数据的过程中，我们发现原始数据本身存在一些噪音，这可能会影响到后续的分析和模型训练。因此，我们采取了一系列的数据清洗操作，以确保数据的准确性和可靠性。首先，我们使用繁体转简体的方式对数据进行了预处理，因为繁体中文与简体中文在书写形式上存在差异，如果不进行转换，可能会导致一些误解。然后，我们进一步对数据进行了清洗，包括清洗特殊字符串、emoji和重复标点等操作。这些操作的目的是去除那些无用的、重复的或者不符合语言规范的信息。在清洗过程中，我们还剔除了一部分无效的样本。无效样本主要是指那些错误句子和正确句子长度差异过大、文本长度过短、中文字符占比低于50%等不符合语言特点的样本。对于部分正确和错误句子对缺失末尾标点的样本，我们统一补上了目标句子末尾可能缺失的标点，以保证数据的完整性。总的来说，我们的数据清洗过程旨在提高数据的质量和可用性，为后续的分析和模型训练提供一个干净、准确的基础。

第二十二届中国计算语言学大会论文集，第213页-第219页，哈尔滨，中国，2023年8月3日至5日。

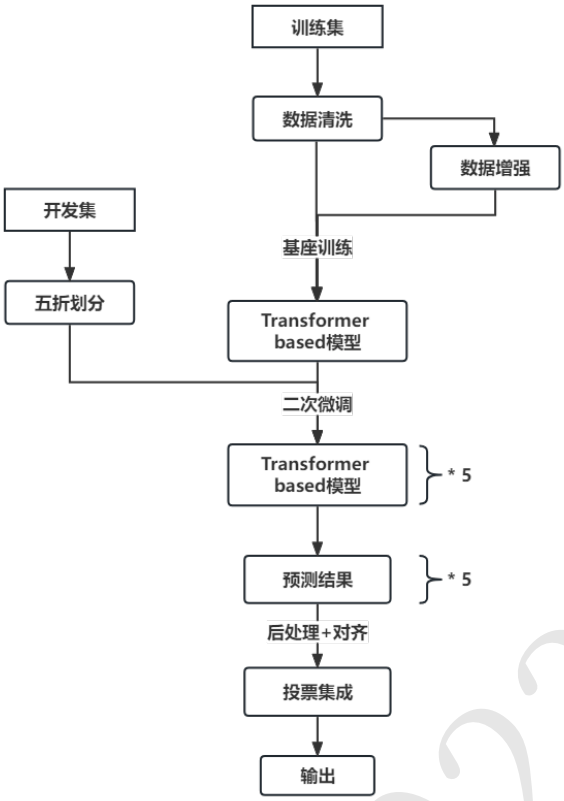

Figure 1: 流程图

4.2 数据增强

我们探讨了如何利用lang8和YACLC数据集来提高自然语言处理模型的性能。鉴于这两个数据集之间的差异，我们在数据增强阶段统计了在YACLC数据集中的各类修改方式所占的比例，并根据占用的比例对lang8的数据进行数据增强。数据增强过程分为三个阶段：

- 数据比例统计：在这个阶段主要分为两部分，首先对YACLC的验证集中的错误类型和修改方式进行统计，计算出在数据中所占有的比例。其次，对lang8和YACLC进行分词，对词频进行统计，并且使用生成的方式，生成一些与高频修改词、易错词的近义词和反义词。

- 单数据增强，在这个阶段会针对每一条数据进行数据多次数据增强。主要的增强方法分成针对结构性错误和语义性错误两种，结构性错误主要是在文字上的重复、丢失、语序变换，语义性的错误包括使用近义词或者反义词对个别词汇进行替换。在实际操作过程中，每一句都会随机出现结构性错和语义性错误，与原来的句子构成句子对，形成新的数据。由于存在两个原则，在最小变更原则下我们适当减少了错误的比例，来实现修改数量的减少，在流利性原则下，增加了语义性的错误以让模型更多的学到语义流利的修改。总而言之，单句数据增强过程中的错误数量和方式参照了YACLC验证集以及最终评价标准。

- 最后本文对大量的单挑增强后数据进行了筛选，主要的目标是够构建一个与YACLC数据集相似的高质量、大数量的训练集。主要方式是对不同类型的数据进行随机抽样，使得比例与YACLC验证集的错误构成一致.

最终，我们将训练集数据和由训练集生成的增强数据进行混合，作为模型第一阶段训练的语料。

4.3 模型设计

基于使用序列到序列模型的考虑，我们主要考虑了两种类型：基于Transformer编码器+解码器的模型和只使用Transformer解码器的生成式模型。这两种模型都涉及到生成阶段，因此使

用Transformer解码器结构几乎是必须的。对于是否使用Transformer编码器结构，我们主要有以下四点考虑。

- Transformer编码器作为语言理解模型，可以对输入的错误句子进行更好地编码和特征提取，以用于正确句子的生成。这是因为Transformer编码器能够捕捉输入句子中的语义信息和上下文关系，从而提高生成正确句子的准确性。例如，在机器翻译任务中，Transformer编码器可以将源语言句子转换为固定长度的向量表示，然后再将这个向量传递给Transformer解码器进行后续的生成过程，中文文本纠错任务与机器翻译任务有着很大的相似性，因此，使用Transformer编码器有着天然的优势。

- 只使用Transformer解码器的生成式模型由于参数量以及预训练数据量堆叠的优势，所涵盖的知识量更加丰富，可以更好地生成正确句子。这是因为Transformer解码器可以直接从输入的错误句子中学习到正确的语法和语义规则，从而生成更加准确的正确句子。同时，在二次微调阶段，我们还可以针对最小改动和流利度提升两个数据集，涉及不同的prompt模板来生成正确的回答，如“请使用尽可能少的改动，修改以下句子<错误句子>，输出：<正确句子>”和“请修改以下句子<错误句子>，使改动后的结果尽可能流畅，输出：<正确句子>”。

- 此外，我们在调研中发现，指针生成网络(Point Generator Network)比较符合本文的特性。指针生成网络旨在用生成式模型解决文本摘要任务，在生成时可以生成新的token，也可以直接复制原文的内容。指针网络可以结合序列到序列模型，并且由于其可以复制原文的特性，这种特性使得指针生成网络在处理文本纠错问题时具有优势(See et al., 2017)。通过复制原文内容，指针生成网络可以避免对原始文本进行过多的修改和调整，从而提高模型的稳定性和可靠性，可以很好地解决模型纠正过多的特性。

- 综上所述，我们在选择序列到序列模型时，综合考虑试验Transformer编码器+解码器的结构模型，Transformer解码器的生成式模型和指针生成网络模型。经过调研，我们选择了官方基线的bart-large，基于bart-large的指针生成网络，和开源的bloom-7b1[0]进行文本纠错方向的实验。

Table 2: 生成式模型prompt模板

	prompt模板
最小改动	请最小变动改正病句：<错误句子>，输出：<正确句子> 我想让你修改病句，使得变动尽可能的小：<错误句子>，输出：<正确句子> 请使用尽可能少的改动，修改以下句子<错误句子>，输出：<正确句子>
流利度提升	我想让你修改病句，达到句子流畅的目的：<错误句子>，输出：<正确句子> 请修改以下句子<错误句子>，使改动后的结果尽可能流畅，输出：<正确句子> 给出一个病句<错误句子>，请你尽可能修改这个句子使其流畅，输出：<正确句子>

在训练阶段，我们首先将训练集和基于训练集增强的数据进行混合，同时输入模型中进行一阶段的训练。在一阶段结束后，我们选取最优的checkpoint作为基座模型，使用开发集进行二阶段微调。我们将最小改动和流利度提升的两个开发集分开进行五折交叉验证和微调，最终在两个任务数据集上各得到五个模型。在使用开源bloom-7b1生成式模型进行训练时，我们设计了多个prompt模板来强化模型的泛化能力，如表格2所示。

4.4 数据后处理

在实验中，我们观察到中文分词器在处理未出现在词表中的词汇时，会生成一个特殊的[UNK]标记。此外，对于一些英文单词，分词器可能会将其拆分成带有的词根形式。这些操作会导致评测过程中出现无意义的编辑操作，从而降低模型的分值。为了解决这个问题，我们对预测输出进行了数据后处理。

[0]https://huggingface.co/bigscience/bloom-7b1

Table 3: 一阶段base模型效果对比

	平均值**F0.5**	最小改动维度**F0.5**	流利提升维度**F0.5**
BART-base	40.59	55.7	25.47
BERT-base	39.85	54.1	25.59
PGN	41.92	57.24	26.59
+data augmentation	44.98	60.26	29.70
bloom-7b1	43.0	58.1	27.89
+data augmentation	44.98	60.4	29.56
BART-large	47.73	62.01	33.45
+data augmentation	**49.69**	**64.06**	**35.32**

- 首先，针对[UNK]标记，我们将原本的错误句子进行再次分词，以识别并还原分词器无法识别的[UNK]标记。这意味着我们会将原本的[UNK]替换回其原始的词汇。这样做可以确保我们的模型能够正确处理所有可能的词汇，从而提高评测结果的准确性。

- 其次，对于带有词根的英文单词，我们会将其拼接成完整的单词，并按照原句格式转换为对应的大小写形式。这样一来，我们的模型就能够正确地处理这些特殊情况，避免因错误的词形变化而导致的编辑操作。

通过这种数据后处理方法，我们成功地提高了模型在评测过程中的表现，使其在面对未出现在词表中的词汇和特殊单词形态时仍能保持较高的准确性。

4.5 模型集成

在五折交叉验证后，我们在每个任务上都得到了五个二次微调后的模型。对于这五个模型的生成结果，我们将其编辑操作提取出来，并设定一个阈值$\theta \in \{1,2,3,4,5\}$对每一个编辑操作进行投票选择。每一个模型的修改方式不同，但是当某一个操作频繁出现时，说明其是“共识”错误。因此，只有当该编辑操作的出现次数大于等于θ时，该操作才会被采纳，这样一来，我们可以筛选出那些更有可能产生积极效果的编辑操作。

5 实验分析

Table 4: 基于bart-large的效果对比

	平均值**F0.5**	最小改动维度**F0.5**	流利提升维度**F0.5**
BART-large	47.73	62.01	33.45
+data augmentation	49.69	64.06	35.32
+second-stage fine-tune	52.23	69.14	35.32
+post-process	54.35	69.11	39.58
+ensemble	**57.6**	**73.05**	**42.15**

接下来，我们对比了不同模型的实验效果。从表3中，我们对比了我们选取的三个模型和对应的使用数据增强方法的实验效果。从表中数据可以看到，我们使用的数据增强方法在三个模型的F0.5得分上，平均能够提升2.33分。同时可以发现，基于bart-large模型的效果好于bloom-7b1和PGN+bart-large。我们初步分析认为原因有三点，1) Transformer的编码器可以在语义理解上为解码阶段提供更多的帮助。2) 基于Transformer解码器的模型可能需要更大的参数量才能发挥效果。3)PGN在生成过程中，过少地改动句子，导致其recall得分较低，因此性能较差。

我们选取了一阶段训练的最好模型，并在此基础上进一步进行二次微调训练。如表4所示，我们使用五折交叉验证划分的开发集二次微调，其中最好的模型能够提高2.54分，在此基础之

上使用数据后处理，可以再次提高2.12分。最终我们使用$\theta = 5$对五个微调模型的编辑操作进行投票，得到57.6分。

6 结论

在本次多维度汉语学习者文本纠错任务中，本队伍使用了设计了基于序列到序列的文本生成式自动化纠错pipeline。数据方面，我们对数据进行清洗并基于开发集数据的错误分布引入了增强数据。模型上，我们分析并试验了基于Transformer编码器+解码器的模型和只基于Transformer解码器的生成式模型，同时对模型生成的结果进行了针对性后处理，并采用投票集成的方式进一步提升性能。最终，我们选取了其中带来有效提升的方法，并最终得到57.6的F0.5得分，位列赛道一封闭任务第三名。

此外，随着大模型的性能日益增强，设计符合任务的prompt是一个值得挑战的方向。例如，给予模型精准的instuction从而挖掘大模型存储的海量知识，或根据模型生成的回答不断给予提示重复迭代以达到最佳的效果，是未来需要进一步思考的方向。

参考文献

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Claudia Leacock, Michael Gamon, and Chris Brockett. 2009. User input and interactions on microsoft research esl assistant. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 73–81.

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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

Alla Rozovskaya and Dan Roth. 2010. Training paradigms for correcting errors in grammar and usage. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 154–162.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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.

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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

第二十二届中国计算语言学大会论文集，第213页-第219页，哈尔滨，中国，2023年8月3日至5日。