

本科毕业设计 (论文)

学院(部)	计算机科学与技术学院		
题　　目	领域词典约束的 文本纠错技术研究与实现		
年　级	19 级	专业	计算机科学与技术（人工智能）
班　级	人工智能 2 班	学号	1928401062
姓　名	张紫岩		
指导老师	李正华	职称	教授
论文提交日期	2023/5/20		

苏州大学
本科毕业设计（论文）独创性声明

本人郑重声明：所提交的本科毕业设计（论文）是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本设计（论文）不含其他个人或集体已经发表或撰写过的研究成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名：张紫岩　　日　期：2023/5/20

苏州大学
本科毕业设计（论文）使用授权声明

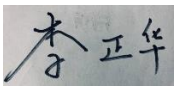
本人完全了解苏州大学关于收集、保存和使用本科毕业设计（论文）的规定，即：本科毕业设计（论文）的著作权以及文中研究成果的知识产权归属苏州大学。苏州大学有权向国家有关部门或第三方机构送交毕业设计（论文）的复印件和电子文档，允许毕业设计（论文）被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编毕业设计（论文），可以将毕业设计（论文）的全部或部分内容编入有关数据库进行检索。

涉密设计（论文）□

本设计（论文）属　　在_____年___月解密后适用本规定。

非涉密设计（论文）□

论文作者签名：日　期：2023/5/20

导 师 签 名：日　期：2023/5/20

领域词典约束的文本纠错技术研究与实现

摘要：

文本纠错是自然语言处理领域的一项重要应用，旨在自动识别并纠正文本中含有的拼写错误、语法错误、语序错误等。文本纠错系统的主要目的是帮助母语写作者更准确地表达观点以及为第二语言学习者提供校对服务。目前，文本纠错技术越来越受到众多研究者的关注，并且已经取得了显著进展。然而，针对特定领域的文本纠错研究还较为空白。

不同于通用领域，特定领域的文本中包含领域内的知识与信息，这些知识对纠错模型的性能有着很大影响，因此，将领域知识引入文本纠错技术中至关重要。领域词典是一种专业化的词典，收录了某个特定领域的术语和概念，反映了该领域的知识结构和规范。构建高质量的领域词典，并将领域词典融入到纠错模型中，可以有效地提高特定领域中文本纠错技术的效果。

针对上述问题，本文首先介绍了文本纠错任务常用的模型和方法。其次，本文面向科技文献写作领域，提出了一种领域词典约束的数据增强方法。最后，本文在爬取的科技文献数据集上进行了实验，分析了不同的数据增强操作对纠错模型性能的影响。

关键字：文本纠错；领域词典；数据增强；序列到序列

Domain dictionary-constrained text correction technology research and implementation

Abstract：

Text correction is an important application of natural language processing (NLP), which aims to automatically identify and correct spelling errors, grammatical errors, word order errors and so on in the text. The main purpose of the text correction system is to help native language writers express their views more accurately and provide proofreading services for second language learners. Currently, text correction technology receives more and more attention from many researchers and has made significant progress. However, research on text correction for specific fields is still relatively blank.

Unlike general fields, texts in specific fields contain knowledge and information within the field, which has a great impact on the performance of the correction model. Therefore, it is crucial to introduce domain knowledge into text correction technology. A domain dictionary is a specialized dictionary that includes terms and concepts from a specific field, reflecting the knowledge structure and norms of that field. Building a high-quality domain dictionary and integrating it into the correction model can effectively improve the effect of text correction technology in specific fields.

Regarding the above problem, this article first introduces the commonly used models and methods for text correction tasks. Secondly, this article proposes a domain dictionary-constrained data augmentation method for scientific literature writing. Finally, this article conducts experiments on the crawled scientific literature dataset to analyze the impact of different data augmentation operations on the performance of the correction model.

Keywords: text correction; domain dictionary; data augmentation; Seq2Seq

目 录

前 言 3

第一章 绪论 5

1.1 课题研究背景 5

1.2 研究现状 5

1.3 论文内容和结构安排 6

第二章 任务描述与数据集介绍 8

2.1 任务定义 8

2.2 两种主流的文本纠错模型 9

2.3.1 序列到序列模型 9

2.3.2 序列到编辑模型 11

2.3 数据集 12

2.3.1 通用数据集 12

2.3.2 领域数据集 12

第三章 针对科技文献领域的文本纠错基准方法 15

3.1 基准模型 15

3.1.1 模型结构 15

3.1.2 目标函数和训练 16

3.1.3 预训练模型 16

3.2 基准实验设置 17

3.2.1 数据集 17

3.2.2 参数设置 17

3.2.3 评价指标 18

3.3 实验结果与分析 19

第四章 领域词典约束的数据增强 21

4.1 领域词典 21
4.2 数据增强技术 21
4.2.1 数据增强技术介绍 21
4.2.2 基于规则的数据增强技术 23
4.3 数据增强实验结果 25
4.3.1 自动评价 25
4.3.2 人工评价 27
第五章 总结与展望 28
5.1 总结 28
5.2 展望 29
参考文献 30
致谢 34

前 言

文本纠错是自然语言处理领域的一项重要应用，旨在自动识别并纠正文本中含有的拼写错误、语法错误、语序错误等。文本纠错技术在输入法、搜索引擎、机器翻译、语音识别等领域都有广泛应用。然而，文本纠错任务本身具有很高的复杂性和难度，需要综合考虑多方面的知识和信息，如字词、句法、语义、上下文、领域等。尤其是在特定领域的文本纠错中，领域知识对文本纠错技术的性能有着很大影响。

领域词典是一种专业化的词典，收录了某个特定领域的术语和概念，反映了该领域的知识结构和规范。领域词典在特定领域的文本纠错任务中具有重要的作用，可以帮助纠错模型识别和纠正领域相关的错误，提高纠错的准确性和可信度。然而，如何有效地利用领域词典进行文本纠错，仍然是一个有待深入研究的问题。目前，针对领域词典约束的文本纠错技术，主要存在以下几个挑战：

(1) 如何构建高质量的领域词典。领域词典的质量直接影响了纠错模型的性能，因此需要对领域词典进行严格的筛选和校验，同时需要及时更新和扩充领域词典，以适应领域知识的变化和发展。

(2) 如何有效地将领域词典融入到纠错模型中，并平衡领域知识和通用知识的权重。领域词典可以提供强约束和指导作用，但也可能导致过拟合或冲突问题。因此需要设计合理的机制和策略，使得纠错模型能够灵活地利用领域词典。

(3) 如何评估领域词典约束的文本纠错技术的效果。由于缺乏统一的评测标准和数据集，目前很难对不同方法进行公平和客观的评价。因此需要构建更细粒度和多维度的评测指标和数据集，以反映不同方法在不同方面的优劣。

针对上述问题，本文从以下几个方面展开研究：

(1) 介绍了文本纠错任务常用的模型和方法，并对比分析了它们优势和劣势。

(2) 面向科技文献写作领域，构建了一个规模较大且包含多种错误类型的数据集，并利用该数据集进行了实验评测。

(3) 介绍了文件纠错任务常用的数据增强技术，构建了一个科技文献领域的领域词典，并提出了一种领域词典约束的数据增强方法。

(4) 在构建的科技文献数据集上进行了数据增强实验，并提出了多维度的评测方法，验证了本文提出的领域词典约束的文本纠错技术的有效性。实验结果表明，本文方法在准确率、召回率和 $F_{0.5}$ 值等指标上表现较好，尤其在领域相关错误的纠正上表现出较大的优势。

第 1 章　绪论

本章主要介绍文本纠错技术的研究背景和研究现状，并阐明了本篇论文的结构和内容。

1.1 课题研究背景

随着信息时代的到来，数据量正在以惊人的速度增长。人类对于自然语言处理的需求也越来越大，机器自动处理数据的方式逐渐取代了人工处理。作为自然语言处理的一项重要应用，文本纠错技术在日常生活中应用广泛，如文本校对工具、写作助手、协助人们进行第二语言的学习等。

早期纠错系统主要采用基于规则的方法，针对特定的语法错误，制定对应的规则进行纠正。这种方法只能纠正特定的几种错误，且纠错效果完全依赖于制定的规则的好坏。为了解决基于规则的纠错方法的局限性，Brockett 等[1] 首先将文本纠错任务看作翻译任务, 并采用了基于统计机器翻译的方法构造纠错模型，通过字、词等相关信息的 n-gram 获取文本特征, 对语言进行建模。但统计方法的泛化能力较差，目前也已经被淘汰。

近年来，基于神经网络的模型由于具有很强的自动特征学习能力，在自然语言处理领域中得到了广泛的应用。许多研究将文本纠错看成一种机器翻译任务，将不符合语法的句子作为源语言，将符合语法的句子作为目标语言。这种方法适用于最先进的神经机器翻译模型。例如，最初被提出的用于机器翻译的编码器-解码器（Encoder- Decoder）模型，已广泛应用于文本纠错，并且效果显著。

1.2 研究现状

在自然语言处理任务中，不符合语法的输入会导致文本的下游解析崩溃，因此从 20 世纪 80 年代开始，研究者就致力于解决“错误输入”的问题。然而，可以协助非母语写作者的有效应用程序直至 2000 年代才开始出现，并且这些程序很大程度上是基于人工编写的“错误规则”。

与此同时，研究人员开始使用有监督的机器学习模型探索更多的数据驱动方法。HOO 共享任务[2] 吸引了 14 个研究小组分享使用英语第一证书语料库（First Certificate in English，FCE）纠正英语限定词和介词错误的实验结果，这标志着文本纠错从基于规则的方法到数据驱动的方法的转折，以及该任务引起了研究者的广泛关注。

接下来的十年见证了更多共享任务的进一步扩展，其中最著名的是 CoNLL-2014[3] 共享任务和 BEA-2019[4] 共享任务，纠错模型的性能也增长了大约三倍。今天，大多数最先进的系统都将该任务视为翻译任务，并使用基于神经机器翻译（Neural Machine Translation，NMT）的模型，即训练模型将错误的源句子翻译成正确的目标句子。然而，这种方法的一个缺点是基于 NMT 的系统需要大量的训练数据才能获得良好的结果，而许多语言的平行数据集的数量很有限。目前最主流的英语纠错模型使用序列到序列的 Transformer[5]，并依赖大量人工数据进行预训练。Grundkiewicz 等[6] 使用基于规则的方法，利用混淆集生成 1 亿个句子对作为预训练数据。Kiyono 等[7] 使用不同的单语种子语料库对几种反向翻译模型变体进行了实验，生成了 7000 万个人工句子对。

中文被认为是世界上最难学的语言之一。与英语不同的是，汉语没有动词时态和复数形式，并且通常有表达同一意思的多种方式。因此，非母语人士在中文写作中出现各种语法错误是很常见的。英语文本纠错领域已经经历了长期的研究并取得了优秀的成果, 但中文纠错历史就要短得多。王洁[8] 利用基于规则的方法，针对介词“比”“把”和“被”字句进行修正，验证了计算机自动修正中文语法错误的可行性。龚小谨等[9] 采用模式匹配的方法和基于句型成分分析的方法进行检查，同时考虑局部和全局的语法限制信息。NLPCC-2018 共享任务[10]是第一个关注这一研究课题的共享任务。在 NLPCC-2018 之前，大多数工作只纠正一种类型的错误，如介词错误 或拼写错误。NLPCC-2018 共享任务的获胜者串联五个不同的模型，并使用 5-gram 语言模型选择最佳输出[11]。

1.3 论文内容和结构安排

首先，本文对目前文本纠错领域的主流模型做了概述，并分析其各自的优缺点。其次，本文介绍了科技文献写作领域数据集的收集以及评价指标，并提出了一种领

域词典约束的数据增强方法，通过分析实验结果可以证明本文提出的数据增强方法的有效性。最后，本文总结了本次实验的不足之处，并提出了下一步研究方向。本文主要分为五章，结构安排如下：

第 1 章：绪论。本章主要阐述文本纠错的研究背景和目的，以及目前课题的研究现状。

第 2 章：任务描述与数据集介绍。本章介绍了文本纠错任务的基本概念、模型概述和数据集。

第 3 章：针对科技文献领域的文本纠错基准方法。本章介绍了基准模型结构和基准模型的实验设置，包括数据集、参数设置、评价指标等，并分析了基准模型实验结果。

第 4 章：领域词典约束的数据增强。本章介绍了领域词典的基本概念和收集方法以及文本纠错领域常用的数据增强技术，并提出了一种领域词典约束的数据增强方法。最后，本章分析了数据增强的实验结果，证实了该方法的有效性。

第 5 章：总结与展望。本章首先对全文进行完善的总结及反思，接着针对一些不足之处进行扩展并提出了一些未来的挑战。

第 2 章　任务描述与数据集介绍

2.1 任务定义

文本纠错任务对文本中的错误进行自动检测和纠正。该任务不仅纠正语法错误，如介词缺失、主谓不一致，还纠正字词和语义错误，如拼写错误和选词错误。

表 2-1：错误类型示例

错误类型	错误句子	正确句子
介词	早饭在餐桌	早饭在餐桌上
拼写	学生大概做飞机去北京。	学生大概坐飞机去北京。
多字	给青年人养成良好习惯也需要时间。	青年人养成良好习惯也需要时间。
语义	我是没有这本书	我没有这本书
标点	明天会下雨吗。	明天会下雨吗？
语序	我的儿子已经大了两个月。	我的儿子已经两个月大了。

表 2-1 展示了几个文本纠错的示例。错误主要可以分为三种类型：替换错误，如第二个句子中“做”被替换为“坐”；省略错误，如第一个句子中介词的省略；插入错误，如第三个句子中多余动词的插入。一个句子中也可能包含多种错误，需要进行一系列的纠正。一些数据集将这些宽泛的类别进一步细分为子类，子类的数量各不相同，具体细节将在 2.3 节介绍。

常规的文本纠错任务的主要目的是帮助母语使用者更准确地表达自己的观点以及为第二语言学习者提供校对服务，所以研究对象以日常写作、第二语言学习者的外语考试为主，目前已取得了显著的进展。然而，针对不同领域的文本纠错研究还较为空白，因此，本文主要面向科技文献领域，探讨科技文献写作中经常出现的错误以及如何利用模型进行自动纠正。

2.2 两种主流的文本纠错模型

文本纠错在神经网络阶段主要采用的模型框架主要可以分为两类，序列到序列（Sequence-to-Sequence，Seq2Seq）模型以及序列到编辑（Sequence-to-Edit，Seq2Edit）模型。

2.2.1 序列到序列模型

随着深度学习的出现，以及机器翻译和其他 Seq2Seq 任务中有前景的结果，NMT 自然被扩展到文本纠错任务。相对于传统的统计机器翻译（Statistical Machine Trans- lation，SMT），NMT 使用单个大型神经网络来模拟整个纠错过程，消除了复杂的文本纠错特定特征工程的需求。尽管 NMT 很简单，但在各种纠错任务上已经实现了最先进的性能。NMT 采用编码器-解码器框架，编码器首先读取整个输入序列 $X = (x_1, \dots, x_T)$ 并将其编码为隐藏状态表示，然后解码器根据输入序列 X 和之前所有生成的单词 $y_1, \dots, y_{t-1}$ 预测下一个单词y_t 来生成输出序列 $y = (y_1, \dots, y_{T'})$：

$$p(y|x) = \prod_{t=1}^{T'} p\left(y_t|y_1, \dots, y_{t-1}, x\right) \qquad (2-1)$$

不同的网络架构被提出用于构建编码器和解码器。三种常见的 Seq2Seq 模型是 RNN[12]、CNN[13] 和 Transformer[9]。

RNN 递归神经网络（Recurrent Neural Network，RNN）是一种专门用于处理序列数据的神经网络，其结构如图 2-1 所示。一个典型的 RNN 网络架构由输入层、隐藏层和输出层组成。为了处理长期依赖关系，门控单元通常用于 RNN。两个最经典的 RNN 模型是长短期记忆网络 (Long Short-Term Memory, LSTM) [14] 模型和门控循环单元（Gated Recurrent Units，GRU）[15] 模型。RNN 在这三个 Seq2Seq 模型中最早被应用在文本纠错任务。Bahdanau 等[12] 引入了一种注意力机制来实现变长表示，从而降低了优化难度并提高了性能。Yuan 和 Briscoe[16] 首次提出了基于 NMT 的文本纠错方法。该模型由一个双向 RNN 编码器和一个基于注意力的 RNN 解码器组成。Xie 等[17] 在文本纠错中使用字符级 RNN 模型。谭咏梅等[18] 针对冠词和介词错误提出了一种基于 LSTM 的序列标注文本纠错方法。

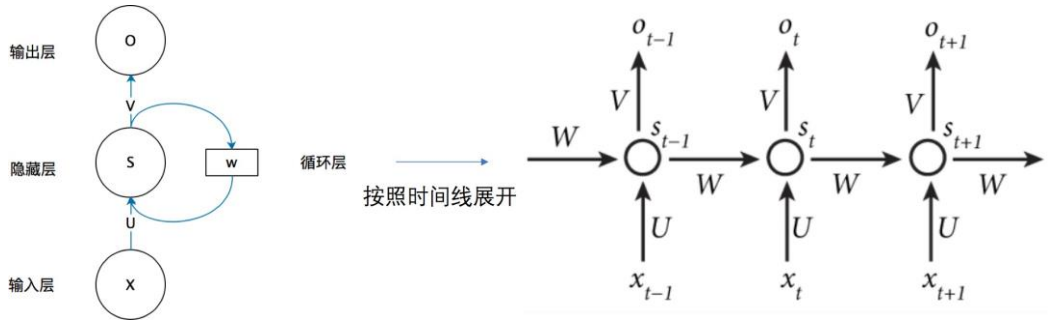


图 2-1: 循环神经网络结构，图片参考文献[12]

CNN 处理序列数据的另一种模型是卷积神经网络 (Convolutional Neural Network, CNN)，其结构如图 2-2 所示。CNN 由卷积层、汇聚层和全连接层交叉堆叠。它可以有效地利用卷积操作来捕捉局部信息，利用池化操作来降低数据的空间复杂度，利用激活函数来增加非线性，利用全连接层输出最终结果。Kalchbrenner 和 Blunsom[19] 首先将 CNN 应用于 NMT，但直到 Gehring 等[20] 将几个 CNN 层堆叠在一起，它们才像 RNN 那样成功。受此工作的启发，Chollampatt 和 Ng[21] 提出了一种用于文本纠错任务的 7 层 CNN 模型。在他们的模型中，局部上下文由在较小的窗口上执行的卷积操作捕获，而更广泛的上下文则由多层结构捕获。这是第一个基于 NMT 的模型，其性能显著优于之前的基于 SMT 的模型。

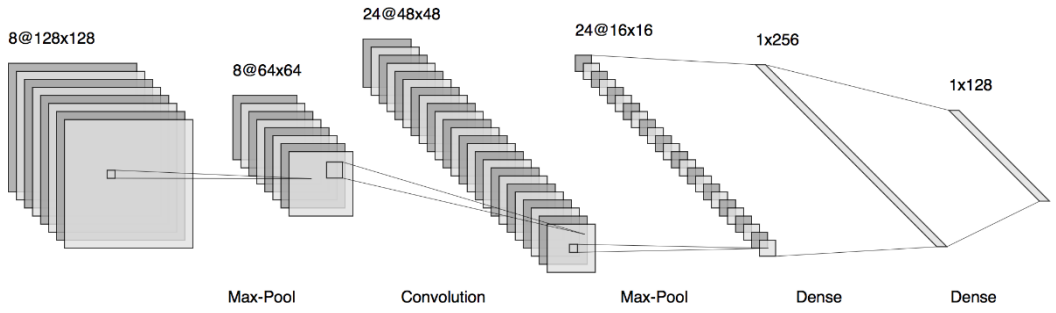


图 2-2: 卷积神经网络结构，图片源自文献[13]

Transformer 2017 年，谷歌提出了 Transformer[9] 模型，不同于 CNN 和 RNN，这是第一个完全依赖于自注意机制来计算其输入表示的 Seq2Seq 模型。Transformer 自问世以来就成为机器翻译任务的首选架构。之前的工作已经研究了 NMT 对文本纠错的适应性，如 Junczys 等[22] 使用编辑加权损失优化模型，Zhao 等[23]在模型中添加复制机制，允许模型直接从源语句中复制词元，而源语句在纠错任务中通常与目标语句有大量重叠，复制机制增强的 Transformer 已经成为纠错任务的流行替代架构。对

Transformer 架构的另一个调整是修改解码器中的注意力机制，以接受和充分利用额外的上下文信息。例如，Kaneko 等[24] 将输入句的 BERT 表示作为文本纠错的附加上下文信息，而 Yuan 和 Bryant[25]在文档中添加了之前的句子，Zhang 等[26]在输入句中添加了基于树的句法表示。

由于 Transformer 体系结构具有大量参数，而文本纠错任务的训练数据有限，因此预训练已成为构建纠错系统的标准流程。第一个基于 Transformer 的纠错系统[22]在语言建模任务中对 Transformer 解码器进行了预训练，但此后更常见的是使用合成伪数据进行预训练。BEA-2019 共享任务中排名第一的系统[27]和最先进的纠错系统[28]都使用合成伪数据对 Transformer 模型进行了预训练，但他们选择的合成伪数据的方式不同。

2.2.2 序列到编辑模型

虽然大多数文本纠错方法根据输入句子生成一个更正后的句子，但 Seq2Edit 模型生成一系列编辑，以应用于输入句子, 如表 2-1 所示。由于纠错任务中从输入到输出的过程中大部分词元被直接复制，Stahlberg 和 Kumar[29]认为生成完整序列是浪费的。通过生成编辑操作而不是完整句子，Seq2Edit 模型通常具有更快的推理速度，比生成整个句子的纠错系统快 5 到 10 倍。然而这种方法的一个限制是编辑操作往往是基于词元的，因此有时无法捕获更复杂的多词元流畅性编辑。

表 2-2: 序列到编辑方法中生成的编辑示例

Source	同时	青年人	养成	良好	习惯	需要	很大	时间
Target	同时	青年人	养成	良好	习惯	需要	很长	时间
Edits	KEEP	KEEP	APP	KEEP	KEEP	KEEP	REP	KEEP

相比于 Seq2Seq 模型，Seq2Edit 模型的主要优点是，它们不仅为校正过程增加了透明度和可解释性，而且在推理时间上更快。然而它们的主要缺点是，由于 Seq2Edit 模型解码器的结构较为简单，所以其性能极度依赖于编码器。如果编码器不使用预训练模型的参数初始化，Seq2Edit 模型几乎是无法使用的。此外，它们通常需要人工定义编辑标签集的大小和范围。因此，本文将文本纠错建模为一个 Seq2Seq 任务，并采用 Transformer 模型作为主模型。

2.3 数据集

本节首先介绍了文本纠错任务的通用数据集，接着介绍了科技文献领域数据集的收集与预处理。

2.3.1 通用数据集

相比于自然语言处理领域的其他任务，文本纠错任务的数据集更为稀缺。通用的中文文本纠错数据集主要有 NLPCC-2018、HSK、MuCGEC，如表 2-3 所示。

表 2-3: 中文文本纠错数据集

语料库	用途	文章/段落数量	字符数量	错误种类	来源
NLPCC-2018	Train	717k+	14.1m+	4	网站
	Test	2k+	61.3k+	4	文章写作
HSK	Train	9k+	3.18m+	4	考试
	Test	2k+	1.06m+	4	考试
MuCGEC	Dev	1.1k+	50k++	19	考试
	Test	5.9k+	228k++	19	考试、写作、网站

NLPCC-2018 NLPCC-2018 共享任务[37]是第一个完全使用普通话进行纠错的共享任务，它发布了 717000 个训练句子，这些句子是从 Lang-8 用户提交的一个清理后的子集中提取的。Lang-8 是一个多国语言学习交流的社区，而 Lang-8 数据集包含了来自 Lang-8 网站的非正式句子，每个句子都由非母语的外语学习者编写，并由母语者进行纠错。尽管 Lang-8 是最大的公开可用语料库之一，但它也是最嘈杂的语料库之一，因为更正是由其他用户而不是专业注释者提供的。与 Lang-8 数据集一样，NLPCC-2018 数据集由其他用户提供更正，因此作者的语言水平是未知的。该共享任务的测试数据来自北京大学中文学习语料库，由 2000 个句子组成，由外国大学生编写。

HSK HSK 数据集是北京语言大学构建的动态作文语料库[38]中的数据，包含了汉语水平考试的词汇、语法、听力、阅读等方面的题目和答案。相较于 Lang-8 数据集，HSK 数据集通过组织专门的标注人员标注的方式得到，数据质量较高。

MuCGEC MuCGEC 是一个全新的并且更为健壮的中文文本纠错数据集,它包含 7063 个句子,约 278k 个字符，样本分别来自于 NLPCC-2018 训练集、NLPCC-2018 测试集和 HSK 语料库数据集。所有的句子都被多个注释者标注，但删除了相同的引用。编辑按照 19 种错误类型进行分类，包括 5 种主要错误类型和 14 种次要错误类型。

2.3.2 领域数据集

为了获取科技文献领域的数据集，本文爬取了中国计算机学会（China Computer Federation，CCF）出版的学术论文，包括计算机科学和技术领域的各个方向，如人工智能、计算机视觉、自然语言处理、机器学习等，共得到两百万条句子。并对数据进行预处理和人工加噪。具体操作如下：

(1) 移除数据集中非正文部分，如致谢、参考文献部分。

(2) 移除数据集中的空白行和噪音数据。

(3) 对正确句子进行人工加噪合成平行句子对，如表 2-4 所示。具体来说，对于给定句子中的每个词元，该方法概率地选择以下操作之一：(i)使用掩码，(ii)删除，(iii)插入从正态分布中采样的随机词元，以及(iv)保留原始词元。具体算法在算法 1 中给出。对于每个词元，选择都是基于分类分布。

表 2-4：人工加噪生成的句子示例

原始句子	**决策树**代表的是对象属性与对象值之间的一种映射关系。
使用掩码	<mask>代表的是对象属性与对象值之间的一种映射关系
原始句子	**虚拟现实**具有相对安全和目标可移动的特点
删除	具有相对安全和目标可移动的特点。
原始句子	**特征提取**将一些原始的输入的数据维度减少。
插入	**决策树特征提取**将一些原始的输入的数据维度减少
原始句子	**卡尔曼滤波**是一种高效率的递归滤波器
保留	**卡尔曼滤波**是一种高效率的递归滤波器

算法 1 人工加噪算法

数据: 正确句子 $Y \in \Gamma$

结果: 合成语料库 D_p

$D_p = \{\}$

$\mu = (\mu_{mask}, \mu_{delete}, \mu_{insert}, \mu_{keep}) \sum \mu = 1$

for $Y \in \Gamma$ **do**

  $X = ()$

  for $j \in (1, ..., J)$ **do**

    随机选取一种操作

    if 选取的操作是保持 **then**

      将 y_j 加入 X

    else if 选取的操作是掩码 **then**

      将 $< mask >$ 加入 X

    else if 选取的操作是删除 **then**

      continue

    else

      将 y_j 加入 X

      w 语料库中随机选取一个单词的概率分布

      将 w 加入 X

    end if

  end for

  $D_p = D_p \cup \{(X, Y)\}$

end for

经过处理后，最终获得 339023 对平行数据集，本文从中随机采样 8000 对平行句子作为验证集，剩余部分用于模型训练。

第 3 章 针对科技文献领域的文本纠错基准方法

本章首先介绍了本文采用的基准模型结构。接着介绍了基准模型的实验设置,包括数据集、参数设置和评价指标。最后，分析了基准模型的实验结果。

3.1 基准模型

3.1.1 模型结构

本文将文本纠错建模为 Seq2Seq 任务，并采用 Fairseq[1] 构建的 Transformer 模型。模型结构如图 3-1 所示。

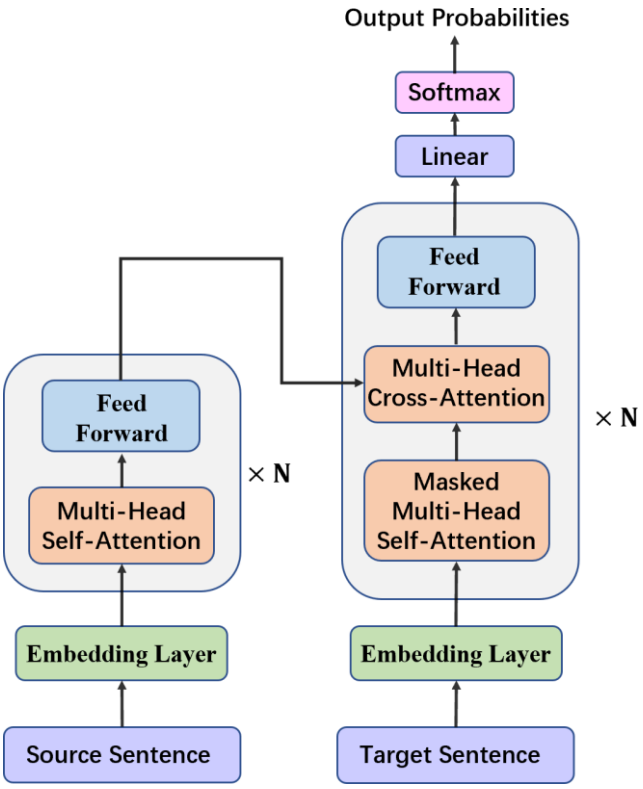


图3-1: 基准模型结构，图片参考文献[9]

给定输入序列$X = (x_1, \ldots, x_T)$，SeqSeq 模型输出序列$Y = (y_1, \ldots, y_T)$。在这个过程中，transformer 模型通过多层编码器计算隐向量表示$H = (h_1, \ldots, h_T)$：

[1]https://github.com/pytorch/fairseq

$$H = Encoder(x_1, \dots, x_n) \quad (3-1)$$

其中编码器的每一层都是一个具有多头注意力机制的 transformer 块。

对输入序列进行编码后，解码器利用输入序列的隐向量预测输出结果。在生成的第 i 步，自注意力解码器以线性的模式预测第 i 个词元 yi，此时解码器状态为：

$$y_i, h_i^d = \text{Decoder}\left(\left[H; h_1^d, \dots, h_{i-1}^d\right], y_{i-1}\right) \quad (3-2)$$

其中解码器的每一层都是包含自注意力机制的 transformer 块。

3.1.2 目标函数和训练

在训练过程中，目标函数是最小化负对数似然损失[30]，正式形式为:

$$\mathcal{L}(\theta) = -\log\left(P(y \mid x; \theta)\right) = -\log \sum_{t=1}^{n} \left(P(y_t \mid y_{<t}; x; \theta)\right) \quad (3-3)$$

其中θ为可训练模型参数，x为源句子，$y = \{y_1, y_2, \dots, y_n\}$代表包含 n 个词元的正确目标句子，$y_{<t} = \{y_1, y_2, \dots, y_{t-1}\}$代表训练时间步长 t 中可见的词元。

在推理过程中，本文利用波束搜索解码[31]通过最大化条件概率$P(y^* \mid x; \theta)$来找到最优序列y^*。

3.1.3 预训练模型

预训练模型是指利用大规模的无标注语料来学习语言的通用表示，然后将这些表示应用到下游的特定任务中，可以有效地学习全局语义表示并明显提升自然语言处理任务的性能。之前的研究表明，预训练模型，例如 BART[32]和 T5[33]，可以大大提高文本纠错任务在从头开始训练后的表现[34]。因此，本文进一步使用 BART 来构建一个更强的基准模型，它与 Transformer 共享相同的模型架构。

本文采用了由 HuggingFace Transformers[35]工具包 [2] 实现的中文 BART[36]，并在 Fairseq 框架中实现它。具体来说，BART 的词汇表中缺失了许多常见的汉字，因此本文从维基百科的语料库中添加了 3866 个汉字和标点符号，在文本纠错数据上对这些新添加的词元进行随机初始化和训练。

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

3.2 基准实验设置

3.2.1 数据集

首先,本文使用中文文本纠错领域的通用数据集 NLPCC-2018、HSK 和 MuCGEC 预训练基准模型。具体来说，本文使用 NLPCC-2018 和 HSK 作为训练集，MuCGEC 验证集作为验证集，并使用 MuCGEC 测试集作为测试集。各个数据集的统计信息如表 3-1 所示。

表 3-1：通用数据集统计信息

数据集	句子数量	错误(%)	用途
NLPCC-2018	1,220,906	89.5	训练
HSK	15,6870	60.8	训练
MuCGEC-dev	1,125	95.1	验证
MuCGEC-test	5,938	92.2	测试

接着，本文使用爬取的领域内数据集对模型进行微调。其中测试集由 500 条人工标注的科技文献领域的平行句子对和 500 条对 CCF 文献数据进行增强得到的伪数据得到，具体数据增强细节将在下一章讨论。各个数据集的统计信息如表 3-2 所示。

3.2.2 参数设置

本文基于 3.1 节所提出的模型架构来设置基准模型：

(1) 本实验采用 Fairseq 构建主干模型，基于 2080Ti 显卡进行模型训练和性能评估。基准模型采用 6 层编码器、6 层解码器架构，即 N=6。

(2) 使用通用数据集训练时，模型采用的学习率为 5e-05。后续采用领域内数据集进行微调时，模型采用的学习率为 1e-05，优化器都采用 Adam。

(3) 模型训练时，每一批次的数据包含 128 个平行句对，允许句子的最大长度为 200 个字，超过此长度的部分将被截断。

(4) 在本文进行的所有 M2 评分实验中，均采用 β=0.5 的设置。

表 3-2：领域数据集统计信息

用途	句子数量	错误(%)	错误种类
训练集	331,023	89.5	训练
验证集	8,000	95.1	验证
测试集	1,000	92.2	测试

3.2.3 评价指标

当前文本纠错研究中使用的最流行的评估方法之一是计算 F_β 分数的 MaxMatch（M2）[39]评分器。具体来说，M^2 评分器是一个基于参考的指标，它将系统假设编辑与人工注释的参考编辑进行比较，如果假设编辑与参考编辑匹配，则计数一次真阳性（TP），如果假设编辑与任何参考编辑都不匹配，则计数一次假阳性（FP），如果参考编辑与任何假设编辑不匹配，则计数一次假阴性（FN）。表 3-3 是每种情况的示例。

一个数据集 TP、FP、FN 的总数可以用来计算准确率（P）和召回率（R）,分别表示假设编辑正确的比例和假设编辑中参考编辑的比例，进而可用于计算 Fβ 分数。目前文本纠错研究中常见的做法是设置 β=0.5。在这种设置中，准确率的权重是召回率的两倍，因为对于纠错系统来说，通常准确率比必须纠正所有错误更重要。

表3-3:M_2评分器示例

		TP		FN	FP		
Original	I	**likes**	to	**drive**	**a**	bicycle	.
Hypothesis	I	**like**	to	**drive**	**the**	bicycle	.
Reference	I	**like**	to	**ride**	**a**	bicycle	.

$$P = \frac{TP}{TP + FP} \tag{3-4}$$

$$R = \frac{TP}{TP + FN} \tag{3-5}$$

$$F_\beta = (1 + \beta^2) \times \frac{P \times R}{(\beta^2 \times P) + R} \tag{3-6}$$

使用编辑重叠度量性能的一个问题是，定义编辑的方法通常不止一种。例如，编辑[has eating→was eaten]也可以实现为[has→was]和[eating→eaten]。如果假设编辑结合了这两种编辑操作而参考编辑没有，则即使该编辑操作产生了相同的有效更正，仍将不被视为 TP。因此，系统性能的测量其实是不准确的。

M2 计分器的创新之处在于，它在原始文本和系统假设之间使用了 Levenshtein 对齐[40]，动态地探索编辑的不同组合方式，从而使假设编辑最大程度地匹配参考编辑。因此，它克服了 HOO 共享任务中使用的评分器可能返回错误分数的限制。当一个测试句子有不止一组参考编辑时，M2 评分器依次尝试每一组参考编辑，并选择针对该测试句子有最佳表现的参考编辑。

3.3 实验结果与分析

首先，本文从头训练一个 transformer 模型并采用上述数据集和实验设置，表 3-4 汇报了基准模型性能。

表 3-4:基准模型性能

模型	准确率(%)	召回率(%)	$F_{0.5}$
文本纠错基准模型	30.0	19.3	27.0

为了进一步验证 BART 预训练模型对基准模型性能的影响，本文加入基于 BART 的预训练模块。表 3-5 汇报了重新实验得到的模型性能。

表3-5:预训练模型有效性实验结果

模型	准确率(%)	召回率(%)	$F_{0.5}$
文本纠错基准模型	30.0	19.3	27.0
加入预训练模块的基准模型	50.9（+20.9）	30.5（+11.2）	44.（+17.9）

上述实验结果表明，加入预训练模型对于模型的性能有显著提升。

第 4 章　领域词典约束的数据增强

本章介绍了文本纠错领域常用的数据增强方法，并面向科技文献领域提出了一种领域词典约束的数据增强方法。实验结果表明，利用该方法生成的伪数据可以使上一章的基准模型的性能有显著提升。

4.1　领域词典

领域词典是一种专业词汇库，包含特定领域的术语、短语和概念，通常用于自然语言处理任务，如文本分类、信息检索和机器翻译等等。领域词典可以帮助计算机更好地理解特定领域的文本，从而提高自然语言处理任务的准确性和效率。领域词典可以通过人工创建或使用自动化工具从大型语料库中提取。人工创建领域词典需要专业知识和经验，通常需要领域专家的帮助。自动化工具可以从大型语料库中提取术语、短语和概念，然后使用统计方法和机器学习算法来确定哪些词汇是最相关的。

文献的关键词是表达文献主题概念的词汇，是从文章的标题、摘要、正文中抽出的，并能表达全文内容主题，具有实在意义的单词或术语。文献的标题和摘要对读者判断文献的参考价值非常重要，而关键词是文献的关键信息和知识点。

因此，本文爬取了训练数据中 CCF 文献的关键词，并进行筛选，保留出现频率高的关键词构成领域词典，最终共得到 23923 个词语。

4.2　数据增强技术

4.2.1　数据增强技术介绍

随着神经网络的不断发展，且模型的尺寸不断增加，人类对高质量数据的需求也在不断增长。为文本纠错模型收集大量训练数据一直是语法纠错领域面临的挑战。文本纠错最大的公开并行数据集 Lang-8 仅包含 200 万个句子对，其中中文语料更是稀缺。可用的数据量不足以支撑模型归纳出的各种语法错误。然而，获取人工标注数据既昂贵又困难，因此，从干净的单语语料库中生成合成平行语料库的技术得到

了广泛的研究。通过在句子中加入杂音，并将其与原句进行配对，生成合成的平行语料库。合适的数据增强方法可以为中文语法纠错模型性能带来稳定的改善。目前，语法纠错数据增强的方法主要有直接噪声和反向翻译噪声。

直接噪声 人工合成语法错误的一种方法是干扰正确文本，使其在语法上不正确。这些干扰可以是基于规则的噪声操作或经常出现在纠错并行语料库中的错误模式。直接噪声所产生的结果在相同的数量级上更容易使用，因为它不需要训练语言模型，也不需要外部数据集。主要可以分为两种类型：基于规则的方法和基于规则的方法。（1）基于规则的方法：向一个干净的语料库添加噪声的最直观的方法是根据一些预定义的规则应用一系列干扰操作，将噪声直接加入到符合语法的句子中。Ehsan 和 Faili[41]从预定义的错误模板中为每个句子添加一个错误，这些错误模板包括省略介词、单词重复等。Lichtarge 等[42]通过删除、插入、替换和交换字符位置，将拼写错误引入维基百科的编辑历史。（2）基于错误模式的方法：生成伪数据的另一种方法是注入文本纠错并行语料库中经常出现的错误。这种方法可以使得合成的错误就更符合自然界中真实错误的发生规律。Rozovskaya 和 Roth[43] 提出了三种不同的文章错误注入方法，即分别根据纠正前文章的分布、纠正后的文章分布，文章被纠正部分的分布来添加错误。Felice 和 Yuan[44] 后来对该方法进行了改进，在产生人工错误时考虑了文本的词法、POS 标签、语义概念和词义信息。Rei[45] 进一步将其扩展到所有类型的误差。

反向翻译噪声 基于反向翻译的方法指的是将训练集中的数据翻转，训练一个从正确句子到错误句子的模型，然后用该模型对大规模单语正确语料加入噪音，再使用人造平行数据对正向模型进行预训练，从而提升模型的鲁棒性。该技术最初被提出是为了在机器翻译中生成额外的数据，但它也可以直接应用于文本纠错。Rei 等[46]首次将反向翻译应用于语法错误检测，Yuan 等[47]在 Rei 等[46]的基础上增加了一种基于语言模型概率的质量控制，以确保生成的合成的伪数据比原始输入句子包含的语法错误更多。Kiyono 等[48]比较了反向翻译噪声和直接噪声，发现反向翻译噪声效果更好。目前，在机器翻译研究领域，由于反向翻译具有较强的性能，被认为是生成伪数据的标准方法。然而，对于文本纠错来说并非如此，单纯地应用反向翻译只会导致性能的微小改善，因为在原始的反向翻译中，模型通常过于保守，只能产生训练集中出现频率较高的错误，也就是说，反向模型同样会受到训练数据不足的影响。

尽管引入预训练语言模型可以一定程度上缓解上述问题，但错误的多样性依旧不足。

4.2.2 基于规则的数据增强技术

综合上述方法的优缺点，本文针对科技文献领域，提出了一种基于规则的数据增强方法，具体操作如下：

混淆词替换 不同于日常生活中的写作，科技文献写作更加书面化，且注重专业词汇的使用。专业词汇是表达学术思想的重要工具，能够准确地传达作者的意图，使读者更好地理解作者的观点。使用专业词汇可以使文献更加准确、规范、严谨，避免了用普通话表达时的歧义和模糊性。因此，本文使用 python 的 synonym[4] 库获取同义词数据，将不在领域词典中的非专业词汇替换领域词典中的专业词汇，用于生成词义混淆错误。

音近字替换 拼写错误在科技文献写作过程中需要着重避免。错别字会影响文章的可读性和可信度，给读者留下不专业、不严谨的印象，导致文章的质量和可信度的降低。此外，错别字还可能导致读者对文章内容的理解产生偏差，甚至误解作者的意图。因此，本文从 CSC2013 中文拼写检查开放任务上获取了汉语音近字 [5]，替换专业词汇中的部分字，形成拼写错误。

删除关键词 在标注数据的过程中，本文发现人们在进行文献写作时常常忽略句子中的主语，造成句子中的主语缺失。科技文献通常需要表达的是一种客观、准确、严谨的语言，缺失主语会导致句子的意思不明确，甚至产生歧义。主语通常是指实验对象、实验方法等专业词汇，缺失主语会使读者难以理解文章的内容和意图，降低文章的质量和可读性。因此，本文删除句子中的专业词汇，通常是位于句首或逗号后充当主语的关键词，造成主语缺失错误。

删除部分关键词 日常生活中，由于专业词汇通常较长且较为拗口，人们经常使用专业词汇的一部分作为简称。但在文献写作中，为了保证文章的准确性和严谨性，应该尽量避免使用简称。如果必须使用简称，也应该在第一次出现时给出全称。因此，针对领域词典内由多个基本词汇组合成的关键词，本文删除其部分，造成关键词省略错误。

每种数据增强操作的示例如表所示。

表 4-1：不同数据增强操作形成的句子示例

原始句子	**决策树**代表的是对象属性与对象值之间的一种映射关系。
混淆词替换	**预测树**代表的是对象属性与对象值之间的一种映射关系
原始句子	**虚拟现实**具有相对安全和目标可移动的特点
删除关键词	具有相对安全和目标可移动的特点。
原始句子	**特征提取**将一些原始的输入的数据维度减少。
音近字替换	**特正提取**将一些原始的输入的数据维度减少
原始句子	**卡尔曼滤波**是一种高效率的递归滤波器
删除部分关键词	**卡尔曼**是一种高效率的递归滤波器

本文提出的数据增强方法完整的流程图如图 4-1 所示。针对一个句子，增强模型首先筛选出句子中的领域词汇，接着针对句子中的领域词汇概率地执行上述四种操作，每种增强操作的概率为 0.25。

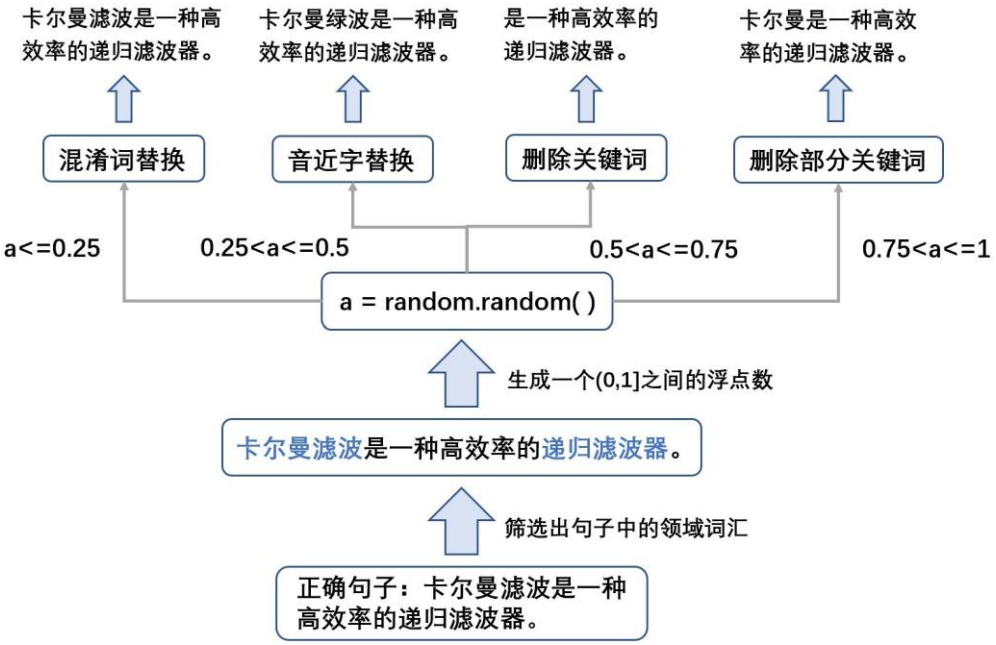


图 4-1: 数据增强流程图

[4]https://pypi.org/project/synonym/

[5]https://github.com/ACL2020SpellGCN/SpellGCN/tree/master/data/gcn_graph.ty_xj

4.3 数据增强实验结果

本文从爬取的 CCF 文献数据集中采样 50 万句，并使用上一小节介绍的基于规则的数据增强方法对它们加入噪音，得到平行语料对基准模型进行微调，实验设置与基准模型实验设置相似。为了证明本文提出的数据增强方法的有效性，实验分别采用自动评价和人工评价两种方式评价数据增强后模型性能的提升。

4.3.1 自动评价

由于科技文献领域真实数据的匮乏，本文采用 500 句人工标注的平行句子对和 500 句人工加噪合成的句子对，加噪方式与基于规则的数据增强方法类似。与基准模型实验一样，本文采用 M 2 评分器作为自动评价的评价指标，实验结果如表 4-2 所示。

表 4-2：数据增强结果

模型	准确率 (%)	召回率 (%)	$F_{0.5}$
去除预训练模块的基准模型	30.0	19.3	27.0
去除预训练模块的数据增强模型	48.5（+18.5）	43.5（+24.2）	47.1（+20.1）
加入预训练模块的基准模型	50.9	30.5	44.9
加入预训练模块的数据增强模型	64.9（+14.0）	50.2（+19.7）	60.9（+16.0）

在对上述结果进行深入分析后，可以得出以下几点结论:

(1) 无论是否加入预训练模块，本文提出的数据增强方法都对基准模型的性能有显著提升，这也从侧面证实了该增强方法的有效性。

(2) 从表中可以看出，当加入预训练模块时，数据增强技术对模型性能的提升比去除预训练模块时要小。这是因为当加入预训练模块时，基准模型的性能较好，因此模型性能的提升空间有限；而去除预训练模块时，基准模型的性能不高，因此有很大的提升空间。

此外，本文还对不同数据增强操作进行了消融实验，证明了它们的有效性，结果如表 4-3 所示。

表 4-3: 数据增强消融实验

加噪类型	准确率 (%)	召回率 (%)	$F_{0.5}$
数据增强模型	63.6	48.5	59.9
混淆词替换	57.3（-7.6）	43.9（-6.3）	54.0（-6.9）
音近字替换	55.6（-9.3）	44.7（-5.5）	53.0（-7.9）
删除关键词	51.9（-13.0）	33.1（-17.1）	46.6（-14.3）
删除部分关键词	52.1（-12.8）	37.9（-12.3）	48.5（-12.4）

随后, 本文对实验结果进行了深入分析, 发现可以总结如下:

(1) 混淆词、音近字替换这两种增强操作形成的错误较为模式化，因此模型能够学习得很好；而另外两种操作对模型的纠正能力的要求更高，所以在消融实验中对模型性能的提升有限。

(2) 当数据为单一句子对时，模型对于缺失关键词这一错误的更正是很困难的。因为模型判断具体缺失了哪个关键词还需要上下文信息，这在只输入单句的情况下是难以做到的。因此，本文将若干个连续的句子合成一个长句子输入模型，性能略有提升。由此可见，开发更复杂的跨句文本纠错模型克服这一困难是有前途的方法。

为了进一步验证本文提出的基于规则的数据增强方法的有效性，本文将基准模型作为反向翻译模型的基本结构，并翻转训练集的方向，对反向翻译模型进行训练。接着，本文使用训练好的反向翻译模型，对大规模单语种子语料库进行加噪，这里使用的种子语料库为爬取的 CCF 文献数据集。最后，本文将基于反向翻译噪声的数据增强模型与基于规则的数据增强模型做对比，结果如表 4-4 所示。

表 4-4: 不同数据增强模型对比实验

加噪类型	准确率 (%)	召回率 (%)	$F_{0.5}$
数据增强模型	64.9	50.2	60.9
反向翻译模型	53.3（-11.3）	35.5（-14.7）	48.4（-12.5）

从对比实验结果中可以看出，本文提出的基于规则的数据增强方法的效果要优于基于反向翻译噪声的数据增强模型。在数据量不足且缺乏高质量数据的情况下，反向翻译模型过于保守，只能产生有限类型的错误，因此对基准模型性能的提升也比较有限。

4.3.2 人工评价

与自动评价不同，针对人工评价方式，本文从 CCF 文献数据集中采样 2 万句进行加噪，形成了一个较大规模测试集。首先，本文使用基准模型和数据增强后的模型在这一测试集上进行预测，并输出两种模型的预测结果。接着，本文筛选出有差异的预测结果，随机采样五百句进行人工评价，在不知道预测结果来自哪一模型的前提下人工确定一个或多个答案。评价结果如表 4-5 所示。

表 4-5: 人工评价结果

模型	预测正确数量	句子总数	准确率 (%)
基准模型	259	500	51.8
数据增强模型	363	500	72.6

经过人工评价发现，数据增强模型的准确率显著优于基准模型。经过进一步分析，本文发现数据增强模型在领域词典相关错误的纠正上表现出较大的优势，相较于基准模型可以更准确地识别错误并纠正。但在非领域内的通用错误上，基准模型和数据增强模型的表现相似。因此，在针对特定领域的文本纠错任务中，平衡领域知识和通用知识的权重对模型性能起着至关重要的作用。

第 5 章　总结与展望

5.1　总结

文本纠错是自然语言处理中的一项重要任务，面向科技文献领域进行文本纠错研究可以提高文献写作的质量，减少读者在阅读中产生的困惑。本文的主要工作及贡献如下：

(1) 总结文本纠错任务常用的模型和方法，以及目前该领域所取得的进展。

(2) 面向科技文献领域，构建了一个规模较大且包含多种错误类型的数据集，补充了该领域数据集的空白，并使用该数据集进行实验评测。

(3) 总结了文本纠错任务常用的数据增强方法。构建了一个领域词典，并提出了一种领域词典约束的数据增强方法，使得模型性能有显著提升。

(4) 采用自动评价和人工评价两种评价方式，进一步验证本文提出的数据增强方法的有效性。实验结果表明，该方法可以使得模型在领域相关错误的纠正上表现出较大优势。

但是，本文工作也存在着以下不足之处：

(1) 相比于通用领域，特定领域的数据集更为稀缺。本文采用通用领域常见的加噪方式对爬取的科技文献句子进行加噪，但这种方式默认发表的科技文献中不包含语法错误，且只关注了文本纠错领域的共性，而忽略了科技文献领域的特性。如何生成科技文献领域的高质量平行语料库，还需进一步研究。

(2) 本文提出的数据增强方法重点关注与领域词典内的词汇相关的错误，但如何平衡领域知识和通用知识的权重，防止模型过度拟合领域错误而忽略了通用错误，还需进一步设计合理的机制和策略。

5.2 展望

尽管过去的十几年中文本纠错领域已经取得了很大进展，但仍面临一些重要挑战。本节为未来的工作提出以下建议：

第一，相比于通用领域，特定领域的数据集更为稀缺。而人工标注数据代价太大，如何自动生成高质量数据势必是一个重点研究方向。

第二，在面向特定领域的文本纠错任务中，将领域知识融入纠错模型需要设计合理的机制和策略，并巧妙平衡领域知识和通用知识的占比，防止产生过拟合或冲突问题。

第三，特定领域的错误更复杂，且不同领域之间的错误分布相差较大。如何构建合理的错误分类体系，或提高现有的错误分类方法的鲁棒性，还需进一步研究。

参考文献

[1] Brockett C, Dolan W B, Gamon M. Correcting ESL errors using phrasal SMT techniques[C]. In Proceedings of ACL, 2006: 249-256.

[2] Dale, Robert, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task[C]. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 54–62.

[3] H T Ng, S M Wu, T Briscoe, et al. The CoNLL-2014 shared task on grammatical error correction[C]. In Proceedings of CoNLL. 2014: 1-14.

[4] Bryant C, Felice M, Andersen Ø E, et al. The BEA-2019 shared task on grammat- ical error correction[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2019: 52-7.

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[6] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 252–263.

[7] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction[C]. In Proceedings of EMNLP-IJCNLP, pages 1236–1242.

[8] 王洁. 计算机识别汉语语法偏误的可行性分析[J]. 语言文字应用,2011(1):135-142.

[9] 龚小谨, 罗振声, 骆卫华. 中文文本自动校对中的语法错误检查 [J]. 计算机工程与应用,2003,39(8):98-100.

[10] Zhao Y, Jiang N, Sun W, et al. Overview of the nlpcc 2018 shared task: Grammaticalmerror correction[C]. In CCF International Conference on NLPCC. Springer, Cham, 2018: 439-445.

[11] Kai Fu, Jin Huang, and Yitao Duan. 2018. Youdao's winning solution to the nlpcc-2018 task 2 challenge: a neural machine translation approach to chinese grammatical error correction[C]. In CCF International Conference on NLPCC, pages 341–350. Springer.

[12] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate[C]. In Proceedings of the 3rd International Conference on Learning Representations.

[13] Gehring, Jonas, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation[C]. In Proceedings of ACL, pages 123–135.

[14] Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long Short-Term Memory[J].Neural Computation, 9(8):1735–1780.

[15] Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[C]. In Proceedings of EMNLP, pages 1724–1734.

[16] Yuan, Zheng and Ted Briscoe. 2016. Grammatical error correction using neural machine translation[C]. In Proceedings of ACL, pages 380–386.

[17] Xie, Ziang, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016.

Neural Language Correction with Character-Based Attention[J]. arXiv: 1603.09727.
[18] 谭咏梅, 杨一枭, 杨林等. 基于 LSTM 和 N-gram 的 ESL 文章的语法错误自动纠正方法 [J]. 中文信息学报,2018,32(06):24-32.
[19] Kalchbrenner, Nal and Phil Blunsom. 2013. Recurrent continuous translation models[C]. Proceedings of EMNLP, pages 8127–8131.
[20] Gehring, Jonas, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation[C]. In Proceedings of ACL, pages 123–135.
[21] Chollampatt, Shamil and Hwee Tou Ng. 2018a. A multilayer convolutional encoder-decoder neural network for grammatical error correction[C]. In Proceed- ings of AAAI, pages 5755–5762.
[22] Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task[C]. In Proceedings of ACL, pages 595–606.
[23] Yuan, Zheng, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 228–239.
[24] Kaneko, Masahiro, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language mod- els in grammatical error correction[C]. In Proceedings of ACL, pages 4248–4254.
[25] Yuan, Zheng and Christopher Bryant. 2021. Document-level grammatical error correction[C]. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 75–84.
[26] Zhang, Yue, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022c. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC- oriented parser[C]. In Proceedings of EMNLP, pages 2518–2531.
[27] Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 252–263.
[28] Stahlberg, Felix and Shankar Kumar. 2021. Synthetic data generation for gram- matical error correction with tagged corruption models[C]. In Proceedings of the16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 37–47.
[29] Stahlberg, Felix and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations[C]. In Proceedings of EMNLP, pages 5147–5159.
[30] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks[J]. Neural computation, 1(2): 270–280.
[31] Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization[C]. In Proceedings of EMNLP, pages 1296–1306.
[32] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, OmerLevy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denois- ing sequence-to-sequence pretraining for natural language generation, translation, and comprehension[C]. In Proceedings of ACL, pages 7871–7880.
[33] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer

learning with a unified text-to-text transformer[J]. Journal of Machine Learning Research (JMLR), 21(140):1–67.

[34] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction[C]. In Proceedings of ACL-IJCNLP, pages 702–707.

[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement De- langue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing[C]. In Proceedings of EMNLP, pages 38–45.

[36] 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 Chi- nese language understanding and generation[J]. arXiv preprint arXiv: 2109.05729.

[37] Zhao, Yuanyuan, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction[C]. In CCF International Conference on NLPCC, pages 439–445.

[38] 张宝林.“HSK 动态作文语料库”的标注问题 [C]. 中文电化教学国际研讨会,2006.

[39] Dahlmeier, Daniel and Hwee Tou Ng. 2012b. Better evaluation for grammatical error correction[C]. In Proceedings of ACL, pages 568–572.

[40] Levenshtein, Vladimir I. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals[J]. Soviet Physics Doklady, 10: 707–710.

[41] Ehsan, Nava and Heshaam Faili. 2013. Grammatical and context-sensitive error correction using a statistical machine translation framework[J]. Software: Practice and Experience, 43(2): 187–206.

[42] Lichtarge, Jared, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction[C].In Proceedings of the 2019 Conference of the North American Chapter of ACL, pages 3291–3301.

[43] Rozovskaya, Alla and Dan Roth. 2010b. Training paradigms for correcting errors in grammar and usage[C]. In Proceedings of NAACL, pages 154–162.

[44] Felice, Mariano and Zheng Yuan. 2014a. Generating artificial errors for grammat- ical error correction[C]. In Proceedings of ACL, pages 116–126.

[45] Rei, Marek, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns[C]. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applica- tions, pages 287–292.

[46] Rei, Marek, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns[C]. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applica- tions, pages 287–292.

[47] Yuan, Zheng, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 228–239.

[48] Kiyono, Shun, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction[C]. In Proceedings of EMNLP-IJCNLP, pages 1236–1242.

[49] Awasthi A, Sarawagi S, Goyal R, et al. Parallel Iterative Edit Models for Local Sequence Transduction[C]. In Proceedings of EMNLP-IJCNLP. 2019: 4251-4261.
[50] Stahlberg F, Kumar S. Seq2Edits: Sequence Transduction Using Span-level Edit Operations[C]. In Proceedings of EMNLP. 2020: 2009.11136.

致谢

首先，我要感谢我的指导老师李正华教授，他对我的毕设题目给予了宝贵的指导和建议，不厌其烦地解答我的疑问和困惑，让我受益匪浅。李老师严谨的治学态度、深厚的专业知识、广阔的学术视野深深地影响了我。

其次，我要感谢龚晨老师和周厚全师兄，他们在我的毕设过程中给予了我很多的帮助和支持，对于我毕业设计中存在的问题，他们认真地给予了许多切实具体的修改建议。

再次，我要感谢我的家人和朋友。和朋友们相处的时光让我感到快乐和幸福，真希望我们能做一辈子的好朋友。父母是我最坚强的后盾和最温暖的港湾，他们一直默默地支持我、鼓励我、理解我。

最后，我要感谢所有参与评审和答辩的老师们，感谢他们在百忙之中审阅我的论文，并提出了宝贵的意见。

苏州大学本科生毕业设计（论文）任务书

学院：计算机科学与技术学院

<table>
<tr><td colspan="4">论文（设计）题目：领域词典约束的文本纠错技术研究与实现</td></tr>
<tr><td>指导教师： 李正华</td><td>职称： 教授</td><td colspan="2">类别： 毕业论文</td></tr>
<tr><td>学生：张紫岩</td><td>学号：192840106</td><td colspan="2">论文（设计）类型：应用型</td></tr>
<tr><td>专业：人工智能</td><td>班级：人智 2 班</td><td>是否隶属科研项目</td><td>否</td></tr>
<tr><td colspan="4">1、论文（设计）的主要任务及目标
文本纠错是自然语言处理的一项重要的底层任务。给定一段含有拼写错误、语法错误的文本，文本纠错任务旨在自动识别错误位置并进行纠正。本次毕业设计面向科技文献写作领域，实现领域词典约束下的文本纠错任务。主要任务及目标为：实现科技文献领域的文本纠错技术。挖掘科技文献中出现的高频词汇，构造领域词典。通过领域词典对数据进行增强，研究数据增强前后文本纠错技术性能的变化。</td></tr>
<tr><td colspan="4">2、 论文（设计）的主要内容
（1）挖掘科技文献中出现的高频词汇，构成领域词典。
（2）通过维基百科的重定向功能将领域词典扩展为领域同义词词典。
（3）数据增强。用同义词替换文本中的部分词语，达到数据增强的效果。
（4）采用序列到序列的深度学习模型实现文本纠错技术。
（5）对比数据增强前后文本纠错技术性能的变化，并分析结果。</td></tr>
<tr><td colspan="4">3、论文（设计）的基本要求
实现文本纠错技术，探讨领域词典约束下的数据增强工作对文本纠错技术性能的提升。论文语言流畅、逻辑清晰。</td></tr>
</table>

4、主要参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.
[2] Yuan Z, Briscoe T. Grammatical error correction using neural machine translation[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 380- 38.
[3] Lichtarge J, Alberti C, Kumar S, et al. Weakly supervised grammatical error correction using iterative decoding[J]. arXiv preprint arXiv:1811.01710, 2018.
[4] Zhao W, Wang L, Shen K, et al. Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data[C]//Proceeding of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 156-165.
[5] Zheng B, Che W, Guo J, et al. Chinese grammatical error diagnosis with long short-term memory networks[C]//Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016). 2016: 49-56.
[6] Chollampatt S, Ng H T. A multilayer convolutional encoder-decoder neural network for grammatical error correction[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018: 32(1).
[7] Hinson C, Huang H H, Chen H H. Heterogeneous Recycle Generation for Chinese Grammatical Error Correction[C]//Proceedings of the 28th International Conference on Computational Linguistics. 2020: 2191-2201.
[8] Zhao Z, Wang H. MaskGEC: Improving Neural Grammatical Error Correction via Dynamic Masking[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(01): 1226-1233.

5、进度安排

	论文（设计）各阶段任务	起 止 日 期
1	收集有关资料，了解该主题的相关研究	3 月 1 日----3 月 7 日
2	系统分析、确定模型阶段	3 月 8 日----3 月 14 日
3	程序设计、模型实现阶段	3 月 15 日----4 月 11 日
4	撰写毕业论文阶段	4 月 12 日----5 月 2 日
5	准备答辩阶段	5 月 3 日----5 月 9 日

注：1、此表一式三份，学院(部)、指导教师、学生各一份；
2、类别是指毕业论文或毕业设计，类型指应用型、理论研究型和其他；
3、在指导教师的指导下由学生填写。

Massive Exploration of Pseudo Data for Grammatical Error Correction

Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui

***Abstract*—Collecting a large amount of training data for grammatical error correction (GEC) models has been an ongoing challenge in the field of GEC. Recently, it has become common to use data demanding deep neural models such as an encoder-decoder for GEC; thus, tackling the problem of data collection has become increasingly important. The incorporation of pseudo data in the training of GEC models is one of the main approaches for mitigating the problem of data scarcity. However, a consensus is lacking on experimental configurations, namely, (i) the methods for generating pseudo data, (ii) the seed corpora used as the source of the pseudo data, and (iii) the means of optimizing the model. In this study, these configurations are thoroughly explored through massive amount of experiments, with the aim of providing an improved understanding of pseudo data. Our main experimental finding is that pretraining a model with pseudo data generated by back-translation-based method is the most effective approach. Our findings are supported by the achievement of state-of-the-art performance on multiple benchmark test sets (the CoNLL-2014 test set and the official test set of the BEA-2019 shared task) without requiring any modifications to the model architecture. We also perform an in-depth analysis of our model with respect to the grammatical error type and proficiency level of the text. Finally, we suggest future directions for further improving model performance.**

***Index Terms*—Natural language processing, language generation, grammars and other rewriting systems, machine translation.**

I. Introduction

TO DATE, a number of studies have tackled grammatical error correction (GEC) as a machine translation (MT) task, in which ungrammatical sentences are regarded as the source language and grammatical sentences are regarded as the target language. This approach allows for the use of cutting-edge neural MT models. For example, the encoder-decoder (EncDec) model [1]–[4], which was originally proposed for MT, has been widely applied to GEC with remarkable results [5]–[13].

Manuscript received December 19, 2019; revised May 12, 2020; accepted June 27, 2020. Date of publication July 7, 2020; date of current version July 30, 2020. This work was supported by JSPS KAKENHI under Grant 19H04162. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nancy F. Chen. *(Corresponding author: Shun Kiyono.)*

Shun Kiyono is with the RIKEN Center for Advanced Intelligence Project, Sendai 980-8579, Japan (e-mail: shun.kiyono@riken.jp).

Jun Suzuki is with the Center for Data-driven Science and Artificial Intelligence, Tohoku University, Sendai 980-8576, Japan (e-mail: jun.suzuki@ecei.tohoku.ac.jp).

Tomoya Mizumoto is with the Future Corporation, Shinagawa 141-0032, Japan (e-mail: t.mizumoto.yb@future.co.jp).

Kentaro Inui is with the Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan (e-mail: inui@ecei.tohoku.ac.jp).

Digital Object Identifier 10.1109/TASLP.2020.3007753

However, a challenge in applying EncDec to GEC is that EncDec requires a large amount of training data [14]; however, the largest set of publicly available parallel data (Lang-8) in GEC only contains 2 million sentence pairs [15]. The amount of available data is insufficient for the model to generalize to various grammatical errors. Consequently, there has been much research on methods for augmenting data by incorporating pseudo training data [9]–[13], [16].

When incorporating pseudo data, several decisions must be made regarding the experimental configurations, namely, (i) the method of generating the pseudo data, (ii) the seed corpus for the pseudo data, and (iii) the optimization setting (Section II). However, a consensus on these decisions in the field of GEC has not been reached. For example, [9] found that a variant of the back-translation [17] method (BACKTRANS (NOISY)) outperformed the method of generating pseudo data from raw grammatical sentences (DIRECTNOISE). However, both the current state of the art model [12] and the winner of the BEA-2019 shared task [13], [18] use the DIRECTNOISE-based method.

In this study, we investigate the aforementioned decisions regarding pseudo data, with the aim to provide the research community with an improved understanding of the incorporation of pseudo data. Through massive amount of experiments, we explore and determine appropriate settings for GEC. In addition, we validate the reliability of the proposed settings by evaluating their performance on benchmark datasets. Specifically, without any task-specific techniques or architecture, our off-the-shelf EncDec method outperforms not only all previous single-model results but also all ensemble results, with the exception of the ensemble result by Grundkiewicz *et al.* [13]. By applying additional task-specific techniques, we further improve the model performance and achieved state-of-the-art performance on the CoNLL-2014 test set and the official test set of the BEA-2019 shared task.

It should be noted that a preliminary version of this study was presented at an international conference (EMNLP-IJCNLP 2019) [19]. Our primary extensions of the conference paper include an additional experiment (Secion IV-B), an in-depth analysis of the experiments (Section V) and a comprehensive overview of previous studies on pseudo data for GEC (Section VI). Specifically, our additions are as follows:

i) We conduct an experiment with an additional pseudo data generation method proposed by Grundkiewicz *et al.* [13] (Section IV-B, Table IV).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

ii) We analyze the effect of changing the seed corpus on the performance of the model for each proficiency level (Section V-A2, Fig. 6, Fig. 7).
iii) We analyze the effectiveness of incorporating pseudo data by comparing the performance of baseline, PRETLARGE, and PRETLARGE+SSE models for each grammatical error type (Section V-B1, Fig. 9). Both PRETLARGE and PRETLARGE+SSE are the proposed settings for GEC (Section IV-E).
iv) We analyze the scalability of the performance for each proficiency level to the amount of pseudo data (Section V-B2, Fig. 10).
v) We summarize previous studies, in which methods are developed for generating pseudo data. These methods include both a rule/probability-based approach and a machine translation-based approach (Section VI-A).
vi) We demonstrate that previous studies used various type of seed corpora (Section VI-B).
vii) We summarize the use of pseudo data for model training in previous studies (Section VI-C).

In addition, a number of revisions have been made, as follows:

i) We ensure the reproducibility of our experiments by publicly releasing the pretrained model files and model outputs.[1]
ii) Our conference paper [19] used a randomly subsampled version of BEA-valid as the validation set. In this study, we instead use the entire BEA-valid and evaluated our models. We confirm that our conclusions about the appropriate settings for GEC remain unchanged.
iii) Part of the experiments (Table VI and Fig. 4) are reran using Gigaword instead of Wikipedia as the seed corpus because we determine that Gigaword was empirically the best seed corpus (Table V).
iv) We add several mathematical equations in the problem formulation (Section II) to describe the GEC task more concisely.

II. PROBLEM FORMULATION AND NOTATION

In this section, we formally define the GEC task addressed in this paper. Let $\mathcal{D}$ be the GEC training data that comprise pairs of an ungrammatical source sentence $\boldsymbol{X}$ and grammatical target sentence $\boldsymbol{Y}$ (i.e., $\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{Y}_n)\}_n$). Here, $|\mathcal{D}|$ denotes the number of sentence pairs in the dataset $\mathcal{D}$.

EncDec is currently the dominant approach to the GEC task [6]–[8]. To describe EncDec, we define $\boldsymbol{X}$ as consisting of a sequence of I tokens, namely, $\boldsymbol{X} = (x_1, \ldots, x_I)$, where x_i denotes the i-th token of $\boldsymbol{X}$. Similarly, y_j denotes the j-th token of $\boldsymbol{Y}$. We define $\boldsymbol{Y}$ as always containing two additional special tokens; $\langle bos \rangle$ for y_0 and $\langle eos \rangle$ for y_{J+1}. Thus, $\boldsymbol{Y} = (y_0, y_1, \ldots, y_J, y_{J+1})$, that is, the length of $\boldsymbol{Y}$ is always $J + 2$. Then EncDec models the following conditional probability:

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{j=1}^{J+1} p(y_j|y_{0:j-1}, \boldsymbol{X}, \boldsymbol{\Theta}), \tag{1}$$

[1][Online]. Available: https://github.com/butsugiri/gec-pseudodata

where $\boldsymbol{\Theta}$ represent all trainable parameters of the model. Our objective is to find the optimal parameter set $\widehat{\boldsymbol{\Theta}}$ that minimizes the following objective function $\mathcal{L}(\mathcal{D}, \boldsymbol{\Theta})$ for the given training data $\mathcal{D}$:

$$\mathcal{L}(\mathcal{D}, \boldsymbol{\Theta}) = -\frac{1}{|\mathcal{D}|} \sum_{(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{D}} \log(p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta})), \tag{2}$$

where $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\Theta})$ denotes the conditional probability of $\boldsymbol{Y}$ given $\boldsymbol{X}$.

In a standard supervised learning setting, the parallel dataset $\mathcal{D}$ comprise only "genuine" parallel data $\mathcal{D}_g$ (i.e., $\mathcal{D} = \mathcal{D}_g$). However, in GEC, it is common to incorporate pseudo data $\mathcal{D}_p$ that are generated from grammatical sentences $\boldsymbol{Y} \in \mathcal{T}$, where $\mathcal{T}$ represents a *seed corpus* (i.e., set of grammatical sentences) [9], [12], [13].

Our interest lies in the following three nontrivial aspects of (2). **Aspect (i)**: There are multiple methods for generating pseudo data $\mathcal{D}_p$ (Section III). **Aspect (ii)**: Options for the seed corpus $\mathcal{T}$ are numerous. To the best of our knowledge, the effect of th seed corpus domain on model performance is yet to be shown. We compare three corpora, namely, Wikipedia, Simple Wikipedia (SimpleWiki) and English Gigaword, as a first trial. Wikipedia and SimpleWiki have similar domains, but different grammatical complexities. Therefore, we can investigate how grammatical complexity affects model performance by comparing these two corpora. We assume that Gigaword contains the smallest amount of noise among the three corpora. We can therefore use Gigaword to investigate whether clean text improves model performance. **Aspect (iii)**: There are at least two major settings for incorporating $\mathcal{D}_p$ into the optimization of Equation (2). One is to use two datasets jointly by concatenating them as $\mathcal{D} = \mathcal{D}_g \cup \mathcal{D}_p$, and then solve the following minimization problem:

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \{\mathcal{L}(\mathcal{D}_g \cup \mathcal{D}_p, \boldsymbol{\Theta})\}. \tag{3}$$

We hereinafter refer to this optimization to as JOINT.

The other setting is to use $\mathcal{D}_p$ for pretraining, namely, minimizing $\mathcal{L}(\mathcal{D}_p, \boldsymbol{\Theta})$ to acquire $\boldsymbol{\Theta}'$, and then fine-tuning the model by minimizing $\mathcal{L}(\mathcal{D}_g, \boldsymbol{\Theta}')$. Specifically, the optimization operates as follows:

$$\boldsymbol{\Theta}' = \arg\min_{\boldsymbol{\Theta}} \{\mathcal{L}(\mathcal{D}_p, \boldsymbol{\Theta})\} \tag{4}$$

$$\widehat{\boldsymbol{\Theta}} = \arg\min_{\boldsymbol{\Theta}} \{\mathcal{L}(\mathcal{D}_g, \boldsymbol{\Theta}')\}. \tag{5}$$

We refer to this optimization as PRETRAIN. We investigate the aforementioned aspects through extensive experiments (Section IV).

III. METHODS FOR GENERATING PSEUDO DATA

In this section, we describe three methods for generating pseudo data: noisy back-translation (BACKTRANS (NOISY)), direct noizing (DIRECTNOISE), and its variant (DIRECTNOISE (SPELL)). In Section IV, we experimentally compare these methods. Examples of each generation method are presented in Fig. 1.

Original:	He died there , but the death date is not clear .
BACKTRANS (NOISY):	He died at there , but death date is not clear .
DIRECTNOISE:	⟨mask⟩ ⟨mask⟩ ⟨mask⟩ , 2 but ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ is not ⟨mask⟩ ⟨mask⟩
DIRECTNOISE (SPELL):	H@@ o died there , but the death te is not clear .
Original:	Gre@@ en@@ space Information for G@@ rea@@ ter London .
BACKTRANS (NOISY):	The information for Gre@@ en@@ space information about G@@ rea@@ ter London .
DIRECTNOISE:	⟨mask⟩ ⟨mask⟩ ⟨mask⟩ for ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ ⟨mask⟩
DIRECTNOISE (SPELL):	Gre@@ en@@ space Information op@@ r@@ ts for G@@ rea@@ ter London .
Original:	The cli@@ p is mixed with images of Toronto streets during power failure .
BACKTRANS (NOISY):	The cli@@ p is mix with images of Toronto streets during power failure .
DIRECTNOISE:	The ⟨mask⟩ is mixed ⟨mask⟩ images si@@ of The ⟨mask⟩ streets large ⟨mask⟩ power R@@ failure place ⟨mask⟩
DIRECTNOISE (SPELL):	The V@@ li@@ p is mixed with images of Toronto streets during power failure .
Original:	At the in@@ stitute , she introduced tis@@ sue culture methods that she had learned in the U.@@ S.
BACKTRANS (NOISY):	At in@@ stitute , She introduced tis@@ sue culture method that she learned in U.@@ S.
DIRECTNOISE:	⟨mask⟩ the the ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ tis@@ culture R@@ methods , she P ⟨mask⟩ the s U.@@ ⟨mask⟩
DIRECTNOISE (SPELL):	At the in@@ stitute , she introduced tis@@ sue culture methods that she had l@@ Gen@@ em@@ d the U.@@ S.

Fig. 1. Examples of sentences generated by BACKTRANS (NOISY), DIRECTNOISE and DIRECTNOISE (SPELL) methods.

A. Noisy Back-Translation: BACKTRANS (NOISY)

Back-translation for the EncDec model was originally proposed for MT by Sennrich *et al.* [17]. In back-translation, a reverse model, which generates a source (ungrammatical) sentence $\boldsymbol{X}$ from a given target (grammatical) sentence $\boldsymbol{Y}$, is trained. Then the output of the reverse model is paired with the input and used as pseudo data.

Currently, in the field of MT research, back-translation is considered to be the *de facto* standard method for generating pseudo data due to its strong empirical performance [20], [21]. However, this is not the case for GEC. Xie *et al.* [9] reported that naively applying back-translation leads to only a minor improvement in performance. Xie *et al.* [9] demonstrate that in vanilla back-translation, the reverse model is generally too conservative and does not generate a sufficient quantity of grammatical errors. To overcome this issue, Xie *et al.* [9] proposed a variant of back-translation[2] called BACKTRANS (NOISY). This method adds $r\beta_{\text{random}}$ to the score of each hypothesis in the beam for every time step. Here, scalar noise r is uniformly sampled from the interval $[0, 1]$, and $\beta_{\text{random}} \in \mathbb{R}_{\geq 0}$ is a hyper-parameter that controls the noise scale. If we set $\beta_{\text{random}} = 0$, then BACKTRANS (NOISY) is identical to vanilla back-translation.

B. Direct Noizing: DIRECTNOISE and DIRECTNOISE (SPELL)

Whereas BACKTRANS (NOISY) generates ungrammatical sentences with a reverse model, DIRECTNOISE injects noise "directly" into grammatical sentences [12], [13], [16], [21]. Specifically, for each token in a given sentence, this method probabilistically selects one of the following operations: (i) masking with a placeholder token $\langle mask \rangle$, (ii) deletion, (iii) insertion of a random token sampled from unigram distribution, and (iv) keeping the original token. A detailed algorithm is provided in Algorithm 1. For each token, the selection is made based on the categorical distribution $(\mu_{\text{mask}}, \mu_{\text{deletion}}, \mu_{\text{insertion}}, \mu_{\text{keep}})$. The DIRECTNOISE algorithm is described in Algorithm 1.

Algorithm 1: DIRECTNOISE Algorithm

Data: Grammatical sentence $\boldsymbol{Y} \in \mathcal{T}$
Result: Pseudo Corpus $\mathcal{D}_p$

$\mathcal{D}_p = \{\}$ // create empty set
$\boldsymbol{\mu} = (\mu_{\text{mask}}, \mu_{\text{deletion}}, \mu_{\text{insertion}}, \mu_{\text{keep}})$ s.t. $\Sigma\boldsymbol{\mu}=1$
for $\boldsymbol{Y} \in \mathcal{T}$ **do**
  $\boldsymbol{X} = (\ )$
  for $j \in (1, \ldots, J)$ **do**
    $action \sim Cat(action|\boldsymbol{\mu})$
    if *action is* keep **then**
      append y_j to $\boldsymbol{X}$
    else if *action is* mask **then**
      append ⟨*mask*⟩ to $\boldsymbol{X}$
    else if *action is* deletion **then**
      continue
    else if *action is* insertion **then**
      append y_j to $\boldsymbol{X}$
      $w \sim unigram_distribution(\mathcal{D}_g)$
      append w to $\boldsymbol{X}$
  $\mathcal{D}_p = \mathcal{D}_p \cup \{(\boldsymbol{X}, \boldsymbol{Y})\}$

Recently, Grundkiewicz *et al.* [13] proposed another method for generating pseudo data. The central idea of their method is to use an off-the-shelf spell checker to generate a confusion set for a given word. Then, they probabilistically replace words with ones in the confusion sets. This method can be generally interpreted as a variant of DIRECTNOISE, in which the masking operation is discarded and the replacement operation is adopted. Thus, we refer to the method as DIRECTNOISE (SPELL).

IV. EXPERIMENTS

The goal of our experiments is to investigate **Aspect (i)–(iii)** introduced in Section II. We design our experiments to ensure

[2]referred as "random noizing" in Xie *et al.* [9].

TABLE I
SUMMARY OF DATASETS USED IN OUR EXPERIMENTS. DATASET MARKED WITH "*" IS A SEED CORPUS $\mathcal{T}$

Dataset	#sent (pairs)	#refs.	Split	Scorer
BEA-train	561,410	1	train	-
BEA-valid	4,384	1	valid	ERRANT
CoNLL-2014	1,312	2	test	ERRANT & M^2 scorer
JFLEG	1,951	4	test	GLEU
BEA-test	4,477	5	test	ERRANT
SimpleWiki*	1,369,460	-	-	-
Wikipedia*	145,883,941	-	-	-
Gigaword*	131,864,979	-	-	-

TABLE II
HYPER-PARAMETER FOR JOINT OPTIMIZATION

Model Architecture	Transformer [4] ("big" setting)
Optimizer	Adam [37] ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. [4]
Number of Epochs	40
Dropout	0.3
Stopping Criterion	Train model for 40 epochs. During the training, save model parameter for every epoch. Then take average of last 20 checkpoints.
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Beam Search	Beam size 5 with length-normalization

that the experimental findings are reproducible and generally applicable to GEC [22]. Specifically, the experiments are based on the following two strategies. (i) We use an off-the-shelf EncDec model without any task-specific architecture or techniques. (ii) We perform hyperparameter tuning, evaluation, and comparison for each method and setting on a validation set. In Section IV-E, we summarize our findings and propose appropriate settings. We then evaluate their performance on multiple benchmark test sets.

A. Experimental Configurations

Dataset: The BEA-2019 workshop official dataset [18] is the origin of the training, validation and test data of our experiments. This dataset consists of the following corpora: the First Certificate in English corpus [23], Lang-8 Corpus of Learner English (Lang-8) [15], [24], the National University of Singapore Corpus of Learner English (NUCLE) [25], and W&I+LOCNESS [26], [27].[3] Hereinafter, we refer to the training data as BEA-train.

The BEA-train is tokenized using the `spaCy` tokenizer.[4] Specifically, we use the `en_core_web_sm-2.1.0` model.[5] We remove sentence pairs that have identical source and target sentences from the training set, following Chollampatt and Ng [7]. We then acquire subwords from target sentences through the byte-pair-encoding (BPE) [28] algorithm. We use `subword-nmt` implementation.[6] We apply BPE splitting to both source and target text. The number of merge operations is set to 8,000.

As a seed corpus $\mathcal{T}$, we use SimpleWiki,[7] Wikipedia,[8] or Gigaword.[9] We apply the noizing methods described in Section III to each corpus and generate pseudo data $\mathcal{D}_p$.[10] The characteristics of each dataset are summarized in Table I.

Evaluation: We report results on BEA-valid, the official test set of the BEA-2019 shared task (BEA-test), the CoNLL-2014 test set (CoNLL-2014) [29], and the JFLEG test set (JFLEG) [30]. All reported results (except for ensemble) are the average of five distinct trials using five different random seeds. We report the scores measured by ERRANT [31], [32][11] for BEA-valid, BEA-test, and CoNLL-2014. Because the reference sentences of BEA-test are not publicly available, we evaluate the model outputs on the `CodaLab`[12] platform for BEA-test. We also report results measured by the M^2 scorer [33] on CoNLL-2014 for comparison with the results of previous studies. We use the GLEU metric [34], [35] for JFLEG.

Model: We adopt the *Transformer* EncDec model [4] for each experiment. Specifically, we use the implementation available in the `fairseq` toolkit [36] and "Transformer (big)" settings of Vaswani *et al.* [4], in which both the encoder and decoder have six layers with 16 attention heads in each layer, a word embedding size d_{model} of 1,024, and a feed-forward network size d_{ff} of 4,096. The dropout rate is set to 0.3.

Optimization: We compare two optimization settings, namely, JOINT and PRETRAIN. For the JOINT setting, we optimize the model with Adam [37]. For the PRETRAIN setting, we pretrain the model with Adam and then fine-tune it on BEA-train using Adafactor [38]. The detailed hyperparameters for each setting are provided in Table II and Table III.

B. Aspect (i): Pseudo Data Generation

We compare the effectiveness of the BACKTRANS (NOISY), DIRECTNOISE, and DIRECTNOISE (SPELL) methods for generating pseudo data. To do this, we first investigate the hyperparameters suitable for BACKTRANS (NOISY) and DIRECTNOISE. Then, we make a comparison of BACKTRANS (NOISY), DIRECTNOISE, and DIRECTNOISE (SPELL). It should be noted that throughout this section, we use (i) the JOINT setting and (ii) all of SimpleWiki as the seed corpus $\mathcal{T}$.

As described in Section III-B, DIRECTNOISE contains four hyperparameters: $(\mu_{\text{mask}}, \mu_{\text{deletion}}, \mu_{\text{insertion}}, \mu_{\text{keep}})$. Running a grid search over all of these parameters is computationally expensive: thus, in this study, we exclusively focus on the effect of μ_{mask}. Therefore we deliberately fix $\mu_{\text{keep}} = 0.2$, and use $\mu_{\text{insertion}} = \mu_{\text{deletion}} = (1 - \mu_{\text{keep}} - \mu_{\text{mask}})/2$. The results are

[3] The data are publicly available at [Online]. Available: https://www.cl.cam.ac.uk/research/nl/bea2019st/.

[4] [Online]. Available: https://spacy.io/

[5] [Online]. Available: https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-2.1.0

[6] [Online]. Available: https://github.com/rsennrich/subword-nmt

[7] https://simple.wikipedia.org

[8] We used the `2019-02-25` dump file at [Online]. Available: https://dumps.wikimedia.org/other/cirrussearch/.

[9] We used English Gigaword Fifth Edition (LDC Catalog No.: LDC2011T07).

[10] The original implementation of DIRECTNOISE (SPELL) is not publicly available. Instead, we use an in-house re-implementation of the method with the hyperparameters described in the paper [13].

[11] [Online]. Available: https://github.com/chrisjbryant/errant

[12] [Online]. Available: https://competitions.codalab.org/competitions/20228

TABLE III
HYPER-PARAMETER FOR PRETRAIN OPTIMIZATION

Pretraining	
Model Architecture	Transformer [4] ("big" setting)
Optimizer	Adam [37] ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. [4]
Number of Epochs	10
Dropout	0.3
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Fine-tuning	
Model Architecture	Transformer [4] ("big" setting)
Optimizer	Adafactor [38]
Learning Rate Schedule	Constant learning rate of 3×10^{-5}
Number of Epochs	30
Dropout	0.3
Stopping Criterion	Use the model with the best validation perplexity on BEA-valid
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Beam Search	Beam size 5 with length-normalization

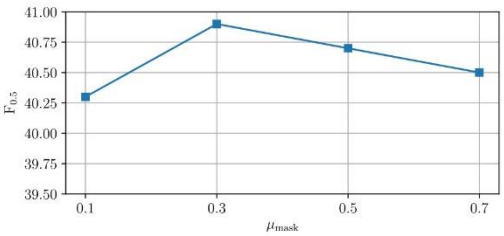


Fig. 2. Performance of the model on BEA-valid with varying parameters DIRECTNOISE (μ_{mask}).

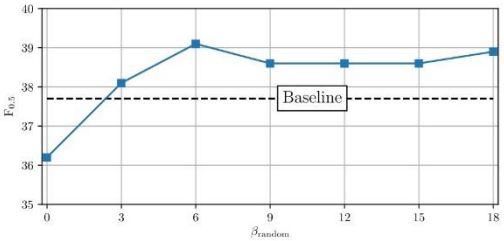


Fig. 3. Performance of the model on BEA-valid with varying parameters BACKTRANS (NOISY) (β_{random}).

summarized in Fig. 2. Here, $\mu_{\text{mask}} = 0.3$ exhibits the best performance; therefore, we use $\mu_{\text{mask}} = 0.3$ for the remainder of the experiments.

We also investigate the effect of varying β_{random} of BACKTRANS (NOISY) by evaluating its performance on BEA-valid Fig. 3). Here, $\beta_{\text{random}} = 6$ achieves the best performance. It

TABLE IV
PERFORMANCE OF MODELS ON BEA-VALID: A VALUE IN **BOLD** INDICATES THE BEST RESULT WITHIN THE COLUMN. THE SEED CORPUS $\mathcal{T}$ IS SIMPLEWIKI

Method	Prec.	Rec.	$F_{0.5}$
Baseline	45.2	22.7	37.7
BACKTRANS (NOISY)	41.7	**31.2**	39.1
DIRECTNOISE	**48.3**	25.4	**40.9**
DIRECTNOISE (SPELL)	44.6	29.5	40.4

should be noted that according to Fig. 3, the performance of back-translation without noise ($\beta_{\text{random}} = 0$) is worse than the baseline. We found that given no noise, the reverse model becomes too conservative, and does not generate grammatical errors, which is consistent with the phenomenon reported by Xie *et al.* [9]. As a result, the pseudo data does not provide useful teaching signals for the model and eventually harm the performance.

Given the suitable hyperparameters for μ_{mask} and β_{random}, we compare the performance of BACKTRANS (NOISY), DIRECTNOISE, and DIRECTNOISE (SPELL). The results are presented in Table IV. The table shows that DIRECTNOISE, DIRECTNOISE (SPELL), and BACKTRANS (NOISY) all achieve better $F_{0.5}$ than the baseline, which is consistent with the result reported by previous studies [9], [12], [13]. It is noteworthy that each method improves different metrics; DIRECTNOISE improves precision, whereas BACKTRANS (NOISY) and DIRECTNOISE (SPELL) improve recall. In addition, DIRECTNOISE achieves superior $F_{0.5}$ to DIRECTNOISE (SPELL). This is surprising because DIRECTNOISE does not rely on the external spell checker. We will exclusively use DIRECTNOISE and BACKTRANS (NOISY) for the rest of the experiments, because they achieve the highest precision and recall, respectively in Table IV.

Why do different methods improve different metrics? We speculate that this result is related to the quality of the language model of EncDec. In the GEC literature, several studies reported that a better language model leads to improved precision. For example, Junczys-Dowmunt and Grundkiewicz [40] incorporated an n-gram language model trained from the Web-scale dataset and improved the precision of the vanilla statistical MT model by almost 10 points. In addition, recently, Chollampatt *et al.* [41] used the scores computed by a pretrained language model (BERT [42]) as features for re-ranking the outputs of the system; they reported an improved precision over the model without BERT. We speculate that a similar effect is occurring in a model trained with DIRECTNOISE, thanks to the existence of the $\langle mask \rangle$ token. Here, the decoder of the model cannot rely on the encoder's hidden states to generate the sentence, because the $\langle mask \rangle$ token removes information from the source sentence. In other words, the decoder is biased toward developing a better language model, rather than merely copying the source sentence.

C. Aspect (ii): Seed Corpus $\mathcal{T}$

We investigate the effectiveness of the seed corpus $\mathcal{T}$ on generating pseudo data $\mathcal{D}_p$. The three corpora (Wikipedia, SimpleWiki and Gigaword) are compared in Table V. We set $|\mathcal{D}_p| =$

TABLE V
PERFORMANCE ON BEA-VALID WHEN CHANGING THE SEED CORPUS $\mathcal{T}$ USED FOR GENERATING PSEUDO DATA ($|\mathcal{D}_p| = 1.4$M)

Method	Seed Corpus $\mathcal{T}$	Prec.	Rec.	$F_{0.5}$
Baseline	N/A	45.2	22.7	37.7
BACKTRANS (NOISY)	Wikipedia	42.8	30.5	39.6
BACKTRANS (NOISY)	SimpleWiki	41.7	31.2	39.1
BACKTRANS (NOISY)	Gigaword	42.2	33.1	40.0
DIRECTNOISE	Wikipedia	47.1	25.8	40.4
DIRECTNOISE	SimpleWiki	48.3	25.4	40.9
DIRECTNOISE	Gigaword	47.3	26.7	41.0

1.4M. The difference in $F_{0.5}$ is small, which implies that the seed corpus $\mathcal{T}$ has only a minor effect on the model performance. Nevertheless, Gigaword consistently outperforms the other two corpora. In particular, DIRECTNOISE with Gigaword achieves the best $F_{0.5}$ value among all configurations. This is a positive result for the GEC community, as Gigaword is a collection of news articles, that can be collected in high quantities at a relatively low cost (e.g., News Crawl[13]).

D. Aspect (iii): Optimization Setting

Here, we compare JOINT and PRETRAIN optimization settings. We are interested in the performance of each setting when the scale of the pseudo data $\mathcal{D}_p$ is (i) approximately the same ($|\mathcal{D}_p| = 1.4$M) and (ii) substantially larger ($|\mathcal{D}_p| = 14$M) than that of genuine parallel data $\mathcal{D}_g$ ($|\mathcal{D}_g| \approx 500$K).

In the case of (ii), we expect that the teaching signal from the pseudo data $\mathcal{D}_p$ becomes dominant in the JOINT setting, and thus, the model may fail to learn from the genuine data $\mathcal{D}_g$. We call this potential problem as **dominant pseudo data**. In order to alleviate this problem, we experiment with upsampling the genuine data $\mathcal{D}_g$ in JOINT, namely, JOINT (UPSAMPLE). Specifically, we search for the appropriate upsampling rate within the values {1, 2, 4, 8, 16, 25} on BEA-valid. Here, 1 is equivalent to JOINT (without upsampling) and 25 approximately corresponds to a ratio of $|\mathcal{D}_p| : |\mathcal{D}_g| = 1 : 1$. As a result, an upsampling rate of 2 achieved the best $F_{0.5}$ on BEA-valid.

1) Joint Training or Pretraining: Table VI presents the results. First, let us compare the result of JOINT and JOINT (UPSAMPLE). In BACKTRANS (NOISY), increasing $|\mathcal{D}_p|$ (1.4M $\rightarrow$ 14M) does not improve $F_{0.5}$ on JOINT (40.0 $\rightarrow$ 40.0). On the other hand, by upsampling the genuine data $\mathcal{D}_g$, JOINT (UPSAMPLE) improves $F_{0.5}$ (40.0 $\rightarrow$ 40.9). These results imply that **dominant pseudo data** indeed exists in vanilla JOINT, and upsampling genuine data can alleviate such a problem.

Second, the table shows that PRETRAIN is superior to JOINT, especially in terms of the properties of *more pseudo data and better performance*. For example, in BACKTRANS (NOISY), increasing $|\mathcal{D}_p|$ (1.4M $\rightarrow$ 14M) improves $F_{0.5}$ on PRETRAIN by more than two points (42.3 $\rightarrow$ 45.6). This is significantly larger than the improvement achieved by JOINT with upsampling (JOINT (UPSAMPLE)); it only improves by 0.9 point (40.0 $\rightarrow$ 40.9). An intuitive explanation for this result is that PRETRAIN effectively handles **dominant pseudo data**, because the model is trained only with $\mathcal{D}_g$ during the fine-tuning phase. Therefore, we conclude that PRETRAIN is the best optimization setting in Table VI.

[13][Online]. Available: http://data.statmt.org/news-crawl/

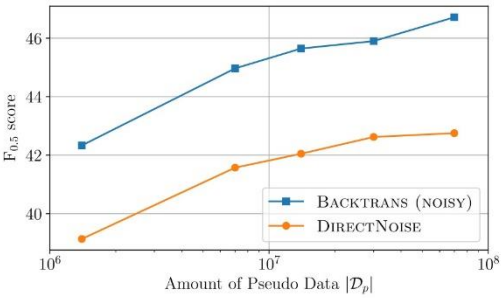


Fig. 4. Performance on BEA-valid for different amounts of pseudo data ($|\mathcal{D}_p|$). The seed corpus $\mathcal{T}$ is Gigaword.

2) Amount of Pseudo Data: We investigate the effect of increasing the amount of pseudo data on the PRETRAIN setting. To do this, we pretrain the model with different amounts of pseudo data {1.4M, 7M, 14M, 30M, 70M}. The results in Fig. 4 demonstrate that the sample efficiency of BACKTRANS (NOISY) is superior to that of DIRECTNOISE. The best model (pretrained with 70M BACKTRANS (NOISY)) achieves $F_{0.5} = 46.7$.

E. Comparison With Current Top Models

The experimental results thus far indicate that the following configurations improve model performance: (i) the combination of JOINT and Gigaword (Section IV-C), (ii) an amount of pseudo data $\mathcal{D}_p$ in JOINT that is not too large (Section IV-D1), and (iii) PRETRAIN with BACKTRANS (NOISY) using a large amount of pseudo data $\mathcal{D}_p$ (Section IV-D2). We summarize these findings and attempt to combine PRETRAIN and JOINT. Specifically, we pretrain the model using 70M pseudo data of BACKTRANS (NOISY). We then fine-tune the model by combining BEA-train and a relatively small amount of DIRECTNOISE pseudo data generated from Gigaword (where $|\mathcal{D}_p| = 250$K). However, the performance does not improve on BEA-valid. Therefore, the best available approach is simply to pretrain the model with a large amount (70M) of BACKTRANS (NOISY) pseudo data and then fine-tune using BEA-train, which we hereinafter refer to as PRETLARGE. We use Gigaword for the seed corpus $\mathcal{T}$ because it has the best performance, as illustrated in Table V.

We evaluate the performance of PRETLARGE on test sets and compared the scores with those of current top models. It is important to note that CoNLL-2014, JFLEG, and BEA-test all involve different domains. For example, CoNLL-2014 consists of essays, while BEA-test contains a much broader type of texts, such as letters, stories, and articles. Therefore, achieving high performance on multiple test sets should ensure that our model's superiority is valid across GEC datasets in general. Table VII reveals that PRETLARGE achieves $F_{0.5} = 61.3$ on

TABLE VI
PERFORMANCE OF THE MODEL WITH DIFFERENT OPTIMIZATION SETTINGS ON BEA-VALID. THE SEED CORPUS $\mathcal{T}$ IS GIGAWORD

Optimization	Method	$\lvert\mathcal{D}_p\rvert$	Prec.	Rec.	$F_{0.5}$
N/A	Baseline	0	45.2	22.7	37.7
PRETRAIN	BACKTRANS (NOISY)	1.4M	49.2	27.2	42.3
PRETRAIN	DIRECTNOISE	1.4M	47.2	23.3	39.1
JOINT	BACKTRANS (NOISY)	1.4M	42.2	33.1	40.0
JOINT	DIRECTNOISE	1.4M	47.3	26.7	41.0
PRETRAIN	BACKTRANS (NOISY)	14M	50.8	32.6	45.6
PRETRAIN	DIRECTNOISE	14M	48.3	27.7	42.0
JOINT	BACKTRANS (NOISY)	14M	40.8	37.1	40.0
JOINT	DIRECTNOISE	14M	48.0	25.0	40.5
JOINT (UPSAMPLE)	BACKTRANS (NOISY)	14M	41.8	37.8	40.9
JOINT (UPSAMPLE)	DIRECTNOISE	14M	47.0	28.0	41.4

TABLE VII
COMPARISON OF OUR BEST MODEL AND CURRENT TOP MODELS: A **BOLD** VALUE INDICATES THE BEST RESULT WITHIN THE COLUMN

		CoNLL-2014 (M^2 scorer)			CoNLL-2014 (ERRANT)			JFLEG	BEA-test (ERRANT)		
Model	Ensemble	Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$	GLEU	Prec.	Rec.	$F_{0.5}$
Chollampatt and Ng (2018) [7]		60.9	23.7	46.4	-	-	-	51.3	-	-	-
Junczys-Dowmunt et al. (2018) [8]		-	-	53.0	-	-	-	57.9	-	-	-
Grundkiewicz and Junczys-Dowmunt (2018) [43]		66.8	34.5	56.3	-	-	-	61.5	-	-	-
Lichtarge et al. (2019) [11]		65.5	37.1	56.8	-	-	-	61.6	-	-	-
Chollampatt and Ng (2018) [7]	✓	65.5	33.1	54.8	-	-	-	57.5	-	-	-
Junczys-Dowmunt et al. (2018) [8]	✓	61.9	40.2	55.8	-	-	-	59.9	-	-	-
Lichtarge et al. (2019) [11]	✓	66.7	43.9	60.4	-	-	-	**63.3**	-	-	-
Zhao et al. (2019) [12]	✓	71.6	38.7	61.2	-	-	-	61.0	-	-	-
Grundkiewicz et al. (2019) [13]	✓	-	-	64.2	-	-	-	61.2	**72.3**	60.1	69.5
PRETLARGE		67.9	44.1	61.3	61.2	42.0	56.0	59.7	65.5	59.4	64.2
PRETLARGE+SSE		68.7	45.2	62.2	62.1	42.7	57.0	60.9	66.1	60.9	65.0
PRETLARGE+SSE+R2L	✓	**72.4**	**46.1**	**65.0**	**67.3**	**44.0**	**60.9**	61.4	72.1	**61.8**	**69.8**

CoNLL-2014, a result that outperforms not only all previous single-model results but also all ensemble results except for that by Grundkiewicz *et al.* [13].

To further improve the performance, we incorporate the following two techniques that are widely used in shared tasks such as BEA-2019 [18] and WMT [44]:

Synthetic Spelling Error (SSE): Lichtarge *et al.* [11] proposed a method of probabilistically injecting character-level noise into a source sentence of pseudo data $\mathcal{D}_p$. Specifically, one of the following operations is applied randomly at a rate of 0.003 per character: deletion, insertion, replacement, or transposition of adjacent characters.

Right-to-Left Re-Ranking (R2L): Incorporating a right-to-left model into the decoding process was independently proposed by Liu *et al.* [45] and Sennrich *et al.* [46], and consistent improvements in performance were reported. Following previous studies [13], [46]–[48], we train four right-to-left models. The ensemble of four left-to-right models generate n-best candidates and their corresponding scores (i.e., conditional probabilities). We then pass each candidate to the ensemble of the four right-to-left models and compute the scores. Finally, we re-rank the original n-best candidates based on the sum of the two scores. We set $n = 5$.

Table VII presents the results of applying SSE and R2L.[14] PRETLARGE+SSE+R2L achieves state-of-the-art performance on both CoNLL-2014 ($F_{0.5} = 65.0$) and BEA-test ($F_{0.5} = 69.8$), which is superior to that of the best system in the BEA-2019 shared task [13].

V. ANALYSIS

In Section IV-E, we demonstrate that PRETLARGE and PRETLARGE+SSE configurations achieve superior performance to that of current top models. In this section, we propose future directions for further improving the performance. We achieve this by analyzing our experimental results in terms of grammatical error type performance and proficiency levels. Specifically, we conduct an analysis on the following two aspects: (i) how the effectiveness of the pseudo data varies across different seed

[14] In the preliminary version of this study [19], we incorporated the technique of Sentence-level Error Detection (SED) [49] in addition to SSE and R2L. SED adopts an external classifier into the evaluation pipeline of the GEC model: the GEC model is applied only if the classifier detects a grammatical error in a given source sentence. While SED improves the performance of the model, it is not directly related to the concept of pseudo data. Thus, we excluded the result incorporating SED from this paper.

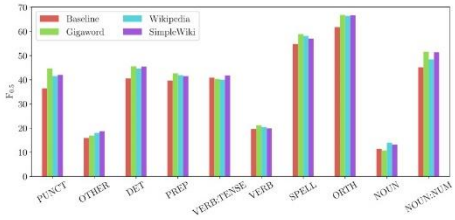


Fig. 5. Performance of the different seed corpora on BEA-valid across various error types. Only the 10 most frequent error types are presented. For a detailed description of each error type, see [31].

corpora (Section V-A), and (ii) the strengths and weaknesses of PRETLARGE (Section V-B).

A. *Effectiveness of Different Seed Corpora*

1) Error Type Analysis: In this section, we analyze the performance of the models trained with different seed corpora through their performance on each error type on BEA-valid.[15] Here, the question is whether one seed corpus is more effective for correcting certain grammatical error types than the other corpora. To do this, we use the DIRECTNOISE models illustrated in Table V. Fig. 5 presents the results. The figure shows that different seed corpora indeed have different characteristics. For example, Gigaword outperforms other seed corpora on error types such as PUNCT, PREP, and SPELL. On the other hand, SimpleWiki outperforms Gigaword on the VERB:TENSE error.

2) Proficiency-Wise Analysis: One notable characteristic of BEA-valid is that it comprises four sections (A, B, C and N) with different English proficiency levels. Here, A (beginner), B (intermediate), and C (advanced) are derived from CEFR levels [50]. The remaining level (N) corresponds to text written by native English speakers. These proficiency levels provide us with increased insight into the behavior of the model. This is because, as illustrated in Fig. 6, the error distribution differs significantly among different proficiency levels. For example, the determiner (DET) error is common in proficiency levels A, B, and C, but not N.

We are interested in the effect of changing the seed corpus on the performance of the model for each proficiency level, as each proficiency level should require a different seed corpus. For example, SimpleWiki should be suitable for proficiency levels A, B, and C because its grammatical complexity is closer to that of English learners. Similarly, Gigaword should be suitable for proficiency level N because its text is written by native English speakers. We analyze this relationship between the seed corpus and proficiency levels using the DIRECTNOISE models illustrated in Table V. Specifically, we evaluated the performance of the model for each proficiency level. Fig. 7 presents the results. The figure indicates that SimpleWiki has either comparative or superior performance to Gigaword for proficiency levels A, B, and C. However, Gigaword outperforms the other two corpora

[15]BEA-valid contains error type annotation for each edit, that is automatically annotated by ERRANT.

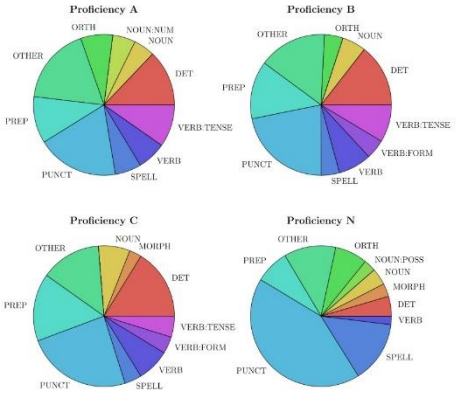


Fig. 6. Error type distribution across different proficiency levels. Only the 10 most frequent error types are presented.

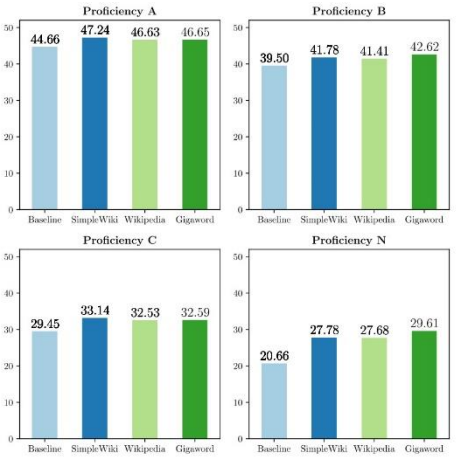


Fig. 7. Effect of the seed corpus on proficiency A, B, B, and N. The y-axis represents the $F_{0.5}$ score.

by almost two points in $F_{0.5}$ score for proficiency level N. These results support our hypothesis that there is a relationship between the grammatical complexity of the seed corpus and the proficiency level.

The fact that a certain seed corpus is more suitable for a certain proficiency level than the other corpora is consistent with our findings in Section V-A1. For example, according to Fig. 6, the VERB:TENSE error is in the 10 most frequent errors in proficiency levels A, B, and C. In Section V-A1, we found that SimpleWiki demonstrates the best performance on the correction of the VERB:TENSE error (Fig. 5). This may be one of the reasons why SimpleWiki shows strong performance on the proficiency levels A, B, and C in Fig. 7.

Source Sentence:	When the concert finished , we went to cloakroom to get signatures from musicians .
Gold Sentence:	When the concert finished , we went to **the** dressing room to get autographs from musicians .
Model Output:	When the concert finished , we went to **the** cloakroom to get signatures from musicians .

Fig. 8. NOUN error generated by our model (PRETLARGE+SSE). **Bold text** indicates the grammatical error successfully corrected by the model. Underlined text indicates the grammatical errors not corrected by the model.

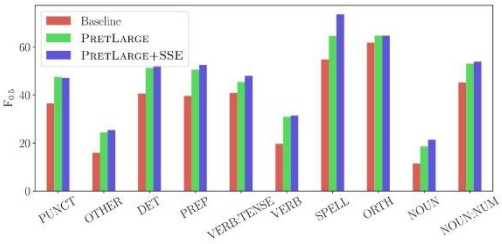


Fig. 9. Performance of the models on BEA-valid across various error types. Only the 10 most frequent error types are presented. For a detailed description of each error type, see [31].

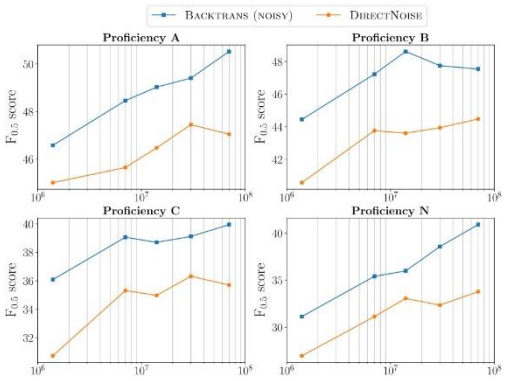


Fig. 10. Performance of the model on each proficiency level in BEA-valid.

B. Strengths and Weaknesses of PRETLARGE

1) Error Type Analysis: We analyze each model through its performance for each error type on BEA-valid (Fig. 9). Specifically, we are interested in the performance of PRETLARGE and PRETLARGE+SSE compared to that of the baseline model, which is only trained with genuine data $\mathcal{D}_g$. It should be noted that BEA-valid contains error type annotation for each edit, that is automatically annotated by ERRANT.

Fig. 9 reveals that PRETLARGE improves the performance across all error types compared to the baseline model. In addition, it is surprising that PRETLARGE+SSE improves the performance of PRETLARGE not only for the SPELL error type but also for most other error types. We speculate that incorporating SSE makes the model more robust against noise.

Fig. 9 also indicates a major weakness of our models, that is, they perform relatively poorly for content word errors, such as NOUN and VERB. This shortcoming is common across GEC models in general; a similar trend is observed in the error type performance of systems participating in the BEA-2019 shared task [18]. One reason for this is that there is an insufficient number of content word errors in both genuine data and pseudo data. For example, as illustrated in Fig. 6, the ratio of NOUN and VERB errors is smaller than the ratio of errors such as PREP, PUNCT, and DET. Thus, developing a method that exclusively generates content word errors is an important direction for future work.

A qualitative analysis through an example in Fig. 8 provides us with other directions for improving the GEC model. Here, the model (PRETLARGE+SSE) successfully corrected the DET error by inserting the missing "the" token. However, the model ignored two NOUN errors in the sentence: one is to replace "cloakroom" with "dressing room", and the other is to replace "signature" with "autograph." For the former error, the model needs to know that the "musicians" are likely to be in the "dressing room" rather than in the "cloakroom." One possible way to do this is to develop a methodology of injecting external commonsense knowledge into the model. For the latter error, the source sentence does not contain enough information for the model to make the correction. This is because "we" in the source sentence cannot be disambiguated: if "we" is the audience, "signature" should be corrected to "autograph." However, supposing that "we" refers to the people from the record label, "signature" seems appropriate (e.g., making a contract). Developing a cross-sentence GEC model [41] is a promising approach to overcome such difficulty of disambiguation.

2) Proficiency-Wise Analysis: We investigated the effect of increasing the amount of pseudo data with respect to each proficiency level. If the performance of a certain proficiency level saturated, then an approach other than increasing the amount of pseudo data would be necessary to improve the performance.

The results are presented in Fig. 10. Here, the performances for proficiency levels A and N scale to the amount of pseudo data, whereas B and C appear to saturate. As discussed in Section V-A2, different seed corpora are appropriate for different proficiency levels. Thus, we may be able to improve the performance of B and C by incorporating a seed corpus with lower grammatical complexity. However, the size of SimpleWiki, which contains only approximately 1.4M sentences, is insufficient for this purpose; extracting text from a raw corpus (e.g., Common Crawl[16]) is thus critical.

[16][Online]. Available: https://commoncrawl.org/

VI. Related Work

A. Methods for Generating Pseudo Data

The lack of genuine data $\mathcal{D}_g$ (i.e., manually annotated error tagged data) has been an ongoing challenge in the field of GEC. The generation of pseudo data has been a central approach for mitigating this problem. There are generally two methods, which can be divided into the following: (i) a rule/probability-based method and (ii) an MT-based method.

A common rule/probability-based method involves applying error templates to external grammatical sentences (i.e., a seed corpus). Here, the templates are generated from a small amount of genuine data. One of the advantages of this method is that a template can be easily manipulated; for example, it can be designed to focus on specific error types, such as mass noun [51], article [52]–[54], and preposition [54]–[56], or it may support grammatical errors in general [16], [57], [58]. More recently, Grundkiewicz *et al.* [13] proposed a similar method; they probabilistically replaced words with ones in confusion sets created by an off-the-shelf English spell checker. Their model achieved the best performance in the BEA 2019 shared task [18].

The another rule/probability-based approach is the one proposed by Zhao *et al.* [12], which is to inject synthetic noise into grammatical sentences. This is heavily inspired by the concept of both denoizing auto-encoders [59] and pretraining gigantic language models [12], [13], [42], [60]. We conducted an experiment using a variant of this method (DirectNoise).

A major disadvantage associated with the rule/probability-based is that it is difficult to obtain high-quality data that closely resembles the errors present in genuine data such as the naturally occurring grammatical error. Several studies have reported that pseudo data may cause performance degradation [57], [58], [61]. Recently, the MT-based method, where a model is used to generate an error from a grammatical sentence, has drawn the attention of the research community. Here, the model is typically either statistical MT [62] or neural MT [9], [11], [63]. For example, Lichtarge *et al.* [11] proposed a round-trip translation. This method regards translation errors generated by the MT model as pseudo grammatical errors. Here, the method is to first translate a given grammatical sentence to an arbitrary bridge language (e.g., Japanese). Then, a sentence is translated back to the original language and used as a source sentence of pseudo data. Another MT-based method is a variant of the back-translation method proposed by Xie *et al.* [9], i.e., Backtrans (noisy) (Section III-A). They demonstrated that Backtrans (noisy) can generate sentences that cannot be distinguished from genuine text containing a grammatical error. In the experiment, we compared Backtrans (noisy) with DirectNoise and demonstrated that the former exhibits superior performance.

Apart from *generating* pseudo data, there exists an approach to *crawl* pseudo data from the Web. Specifically, Lichtarge *et al.* [11] extracted Wikipedia's revision histories and constructed pseudo data, namely, Wikipedia Revisions. This approach is orthogonal to our study; one may jointly use generated pseudo data and Wikipedia Revisions.

B. Seed Corpus

Historically, numerous types of corpora, e.g., collection of news articles [9], [12], [13], [51], [55], British National Corpus [61], and English Wikipedia [53], [55], [56], [58], have been considered to be the seed corpus for generating pseudo data. This fact implies that consensus has not yet been achieved with respect to the choice of seed corpus. Our research question on the effectiveness of seed corpus on the model performance has been formulated based on such a situation.

Felice and Yuan [58] has the motivation similar to our study; the authors denoted that the variables of seed corpus, including the (i) topic (ii) genre (iii) style (iv) text complexity, and (v) native language of the writer, should be considered. However, they only considered English Wikipedia as the seed corpus; and thus, the no sufficient conclusion could be obtained with respect to these variables. In our study, we conducted controlled experiments and compared three distinctive seed corpora (Section IV-C). We found that Gigaword is the best seed corpus, at least in our setting.

C. Optimization Settings

As discussed in Section II, there are at least two means of optimization, i.e., Joint and Pretrain. The effectiveness of Joint optimization has been observed in both non-neural models [57], [58] and neural models [9]. However, the Pretrain approach is being actively explored in the GEC field.

The origin of the Pretrain approach with respect to the GEC model is "partial pretraining" of EncDec. Junczys-Dowmunt *et al.* [8] initialized the embedding matrix and decoder parameters of EncDec with Word2Vec vectors [64] and pretrained language model respectively. Further, they reported that both procedures consistently improved the performance of EncDec. A similar approach has also been used by Chollampatt and Ng [41]. However, more recently, "full pretraining" of EncDec has become dominant [11]–[13], [16], owing to its strong empirical performance. This approach pretrains the entire EncDec using pseudo data, and subsequently fine-tunes it with genuine data.

One interesting aspect of GEC is that even though it incorporates same model (EncDec) as MT, the dominant optimization setting is different. The Joint setting is commonly used in MT [21], [65], whereas Pretrain is used in the existing GEC studies. We compared both settings and confirmed the superiority of Pretrain (Section IV-D).

VII. Conclusion

In this study, we investigated several aspects of the incorporation of pseudo data in GEC. By conducting a massive amount of experiments, the following procedures are concluded to be effective for training the model and for obtaining a strong performance:

i) utilize Gigaword as the seed corpus (Section IV-C);
ii) pretrain the model with large amount (e.g., 70M) of Backtrans (noisy) data (Section IV-D); and
iii) fine-tune the pretrained model using genuine data (Section IV-D).

Further, we demonstrated the effectiveness of this proposal by achieving state-of-the-art performance using the CoNLL-2014 and BEA-2019 test sets (Section IV-E).

Subsequently, we conducted an in-depth analysis of the experimental results; our findings can be summarized as follows.

i) The suitable seed corpus varies across various grammatical error types and proficiency levels: for example, a seed corpus exhibiting low grammatical complexity is suitable for low-proficiency texts (Section V-A1, Section V-A2).
ii) When compared with the baseline, our proposed setting (PRETLARGE) improves the performance with respect to all the grammatical error types. However, the content word errors remain a challenge (Section V-B1).
iii) The performance at each proficiency level shows promising scalability with respect to the amount of pseudo data. However, the performances at proficiency levels B and C seem to saturate when using the largest data (Section V-B2).

Based on these observations, several possible approaches can be suggested to further improve model performance, especially on content word errors. For example, developing (1) a pseudo data generation method that can exclusively generate content word errors, (2) a means of injecting commonsense knowledge into the model, and (3) more sophisticated cross-sentence GEC are promising approaches.

ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers for their insightful comments.

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst. 28*, 2014, pp. 3104–3112.
[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[3] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.
[4] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 31*, 2017, pp. 5998–6008.
[5] Z. Yuan and T. Briscoe, "Grammatical error correction using neural machine translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, 2016, pp. 380–386.
[6] J. Ji, Q. Wang, K. Toutanova, Y. Gong, S. Truong, and J. Gao, "A nested attention neural hybrid model for grammatical error correction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, 2017, pp. 753–762.
[7] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5755–5762.
[8] M. Junczys-Dowmunt, R. Grundkiewicz, S. Guha, and K. Heafield, "Approaching neural grammatical error correction as a low-resource machine translation task," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2018, pp. 595–606.
[9] Z. Xie, G. Genthial, S. Xie, A. Ng, and D. Jurafsky, "Noising and denoising natural language: Diverse backtranslation for grammar correction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2018, pp. 619–628.
[10] T. Ge, F. Wei, and M. Zhou, "Fluency boost learning and inference for neural grammatical error correction," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 1055–1065.
[11] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. Tong, "Corpora generation for grammatical error correction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 3291–3301.
[12] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, "Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 156–165.
[13] R. Grundkiewicz, M. Junczys-Dowmunt, and K. Heafield, "Neural grammatical error correction systems with unsupervised pre-training on synthetic data," in *Proc. 14th Workshop Innovative Use NLP Building Educ. Appl.*, 2019, pp. 252–263.
[14] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proc. 1st Workshop Neural Mach. Transl.*, 2017, pp. 28–39.
[15] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning SNS for automated japanese error correction of second language learners," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, 2011, pp. 147–155.
[16] Y. J. Choe, J. Ham, K. Park, and Y. Yoon, "A neural grammatical error correction system built on better pre-training and sequential transfer learning," in *Proc. 14th Workshop Innovative Use NLP Building Educa. Appl.*, 2019, pp. 213–227.
[17] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, 2016, pp. 86–96.
[18] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in *Proc. 14th Workshop Innovative Use NLP Building Educa. Appl.*, 2019, pp. 52–75.
[19] S. Kiyono, J. Suzuki, M. Mita, T. Mizumoto, and K. Inui, "An empirical study of incorporating pseudo data into grammatical error correction," in *Proc. Conf. Empirical Methods Natural Lang. Process. and 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1236–1242.
[20] B. Haddow *et al.*, "The University of Edinburgh's submissions to the WMT18 news translation task," in *Proc. 3rd Conf. Mach. Transl.*, 2018, pp. 399–409.
[21] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 489–500.
[22] X. Bouthillier, C. Laurent, and P. Vincent, "Unreproducible research is reproducible," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 725–734.
[23] H. Yannakoudakis, T. Briscoe, and B. Medlock, "A new dataset and method for automatically grading ESOL texts," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguist.*, 2011, pp. 180–189.
[24] T. Tajiri, M. Komachi, and Y. Matsumoto, "Tense and aspect error correction for ESL learners using global context," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguist.*, 2012, pp. 198–202.
[25] D. Dahlmeier, H. T. Ng, and S. M. Wu, "Building a large annotated corpus of learner English: The NUS corpus of learner english," in *Proc. 8th Workshop Building Educa. Appl. Using NLP*, 2013, pp. 22–31.
[26] H. Yannakoudakis, Ø. E. Andersen, A. Geranpayeh, T. Briscoe, and D. Nicholls, "Developing an automated writing placement system for ESL learners," *Appl. Meas. Edu.*, vol. 31, no. 3, pp. 251–267, 2018.
[27] S. Granger, "The computer learner corpus: A versatile new source of data for SLA research," in *Learner English on Computer*, S. Granger, Ed. London, U.K. and New York, NY, USA: Addison-Wesley, 1998, pp. 3–18.
[28] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, 2016, pp. 1715–1725.
[29] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The CoNLL-2014 shared task on grammatical error correction," in *Proc. 18th Conf. Comput. Natural Lang. Learn.: Shared Task*, 2014, pp. 1–14.
[30] C. Napoles, K. Sakaguchi, and J. Tetreault, "JFLEG: A fluency corpus and benchmark for grammatical error correction," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2017, pp. 229–234.
[31] C. Bryant, M. Felice, and T. Briscoe, "Automatic annotation and evaluation of error types for grammatical error correction," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.*, 2017, pp. 793–805.
[32] M. Felice, C. Bryant, and T. Briscoe, "Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments," in *Proc. COLING 26th Int. Conf. Comput. Linguist.: Tech. Papers*, 2016, pp. 825–835.
[33] D. Dahlmeier and H. T. Ng, "Better evaluation for grammatical error correction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2012, pp. 568–572.
[34] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "Ground truth for grammatical error correction metrics," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguist. and 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 588–593.

[35] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "GLEU without tuning," 2016, *arXiv:1605.02592*.
[36] M. Ott *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2019, pp. 48–53.
[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[38] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4603–4611.
[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2818–2826.
[40] M. Junczys-Dowmunt and R. Grundkiewicz, "Phrase-based machine translation is state-of-the-art for automatic grammatical error correction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1546–1556.
[41] S. Chollampatt, W. Wang, and H. T. Ng, "Cross-sentence grammatical error correction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, 2019, pp. 435–445.
[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, 2019, pp. 4171–4186.
[43] R. Grundkiewicz and M. Junczys-Dowmunt, "Near human-level performance in grammatical error correction with hybrid machine translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2018, pp. 284–290.
[44] L. Barrault *et al.*, "Findings of the 2019 conference on machine translation (WMT19)," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 1–61.
[45] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Agreement on target-bidirectional neural machine translation," in *Proc. Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2016, pp. 411–416.
[46] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," in *Proc. 1st Conf. Mach. Transl.*, 2016, pp. 371–376.
[47] R. Sennrich *et al.*, "The University of Edinburgh's neural MT systems for WMT17," in *Proc. 2nd Conf. Mach. Transl.*, 2017, pp. 389–399.
[48] M. Morishita, J. Suzuki, and M. Nagata, "NTT neural machine translation systems at WAT 2019," in *Proc. 6th Workshop Asian Transl.*, 2019, pp. 99–105.
[49] H. Asano, M. Mita, T. Mizumoto, and J. Suzuki, "The AIP-Tohoku system at the BEA-2019 shared task," in *Proc. 14th Workshop Innovative Use NLP Building Educa. Appl.*, 2019, pp. 176–182.
[50] D. Little, "The common European framework of reference for languages: Content, purpose, origin, reception and impact," *Lang. Teach.*, vol. 39, no. 3, pp. 167–190, 2006.
[51] C. Brockett, W. B. Dolan, and M. Gamon, "Correcting ESL errors using phrasal SMT techniques," in *Proc. 21st Int. Conf. Comput. Linguist. and 44th Annu. Meeting Assoc. Comput. Linguist.*, 2006, pp. 249–256.
[52] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara, "Automatic error detection in the Japanese Learners' English spoken data," in *Proc. Companion Volume Proc. 41st Annu. Meeting Assoc. Comput. Linguist.*, 2003, pp. 145–148.
[53] A. Rozovskaya and D. Roth, "Training paradigms for correcting errors in grammar and usage," in *Proc. Human Lang. Technol.: Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2010, pp. 154–162.
[54] A. Rozovskaya, M. Sammons, and D. Roth, "The UI system in the HOO 2012 shared task on error correction," in *Proc. 7th Workshop Building Educa. Appl. Using*, 2012, pp. 272–280.
[55] A. Rozovskaya and D. Roth, "Generating confusion sets for context-sensitive error correction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2010, pp. 961–970.
[56] A. Cahill, N. Madnani, J. Tetreault, and D. Napolitano, "Robust systems for preposition error correction using wikipedia revisions," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, 2013, pp. 507–517.
[57] Z. Yuan and M. Felice, "Constrained grammatical error correction using statistical machine translation," in *Proc. 7th Conf. Comput. Natural Lang. Learn.: Shared Task*, 2013, pp. 52–61.
[58] M. Felice and Z. Yuan, "Generating artificial errors for grammatical error correction," in *Proc. Student Res. Workshop 14th Conf. Eur. Chapter Assoc. Comput. Linguist.*, 2014, pp. 116–126.
[59] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[60] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MASS: Masked sequence to sequence pre-training for language generation," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5926–5936.
[61] J. Foster and O. Andersen, "GenERRate: Generating errors for use in grammatical error detection," in *Proc. 4th Workshop Innovative Use NLP Building Educa. Appl.*, 2009, pp. 82–90.
[62] M. Rei, M. Felice, Z. Yuan, and T. Briscoe, "Artificial error generation with machine translation and syntactic patterns," in *Proc. 12th Workshop Innovative Use NLP Building Educ. Appl.*, 2017, pp. 287–292.
[63] S. Kasewa, P. Stenetorp, and S. Riedel, "Wronging a right: Generating better errors to improve grammatical error detection," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4977–4983.
[64] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst. 27*, 2013, pp. 3111–3119.
[65] I. Caswell, C. Chelba, and D. Grangier, "Tagged back-translation," in *Proc. 4th Conf. Mach. Transl.*, 2019, pp. 53–63.



Shun Kiyono received the B.E. degree and the M.Info.Sci. degree from Tohoku University, Japan, in 2017 and 2019, respectively. In 2019, he joined the Center for Advanced Intelligence Project, RIKEN. His research interests include machine learning and natural language processing.



Jun Suzuki received the B.E. and M.Eng. degrees from Keio University, Japan, in 1999 and 2001, respectively and the Ph.D. degree in engineering from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2005. In 2001, he joined NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corp. (NTT). Then, he joined Tohoku University in 2018 as an Associate Professor of the Graduate School of Information Sciences. He is currently a Full Professor of the Center for Data-driven Science and Artificial Intelligence at Tohoku University. His research interests include machine learning and natural language processing.



Tomoya Mizumoto received the B.E. degree from Konan University, Japan, in 2010, and the M.E. and Ph.D. degrees from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2012 and 2015, respectively. He was a Research Fellow (DC2) of the Japan Society for the Promotion of Science from 2013 to 2015. From 2015 to 2017, he was a Post-doctoral Researcher with the Tohoku University, Japan. From 2017 to 2019, he was a Post-doctoral Researcher with the Center for Advanced Intelligence Project, RIKEN. He is currently a Research Engineer with Future Corporation, Japan and a Visiting Researcher with the Center for Advanced Intelligence Project, RIKEN. His research interests include grammatical error correction, short answer scoring and natural language processing.



Kentaro Inui is a Professor of the Graduate School of Information Sciences at Tohoku University, where he is head of the Natural Language Processing Lab. He also leads the Natural Language Understanding Team at the RIKEN Center for the Advanced Intelligence Project. His professional career started as Assistant Professor at Tokyo Institute of Technology. He then joined Kyushu Institute of Technology and Nara Institute of Science and Technology as Associate Professor in 1998 and 2002, respectively, before joining Tohoku University as Professor in 2010. His research areas include natural language processing and artificial intelligence. He currently serves as General Chair of EMNLP-IJCNLP 2019, Director of Association for Natural Language Processing, Member of Science Council of Japan, and Director of NPO FactCheck Initiative Japan.

语法纠错伪数据的大量探索

Shun Kiyono , Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui

IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 2134-2145.

摘要

为语法纠错（Grammatical Error Correction，GEC）模型收集大量训练数据一直是语法纠错领域面临的挑战。最近，对数据要求高的深度学习模型，如编码器-解码器架构在语法纠错领域被广泛运用。因此，解决收集数据这一问题变得越来越重要。在 GEC 模型的训练中引入伪数据是缓解数据稀缺问题的主要方法之一。然而，对于实验配置，即 (i) 生成伪数据的方法，(ii) 作为伪数据来源的种子语料库，(iii) 优化模型的手段，仍旧缺乏共识。在本研究中，为了加强对伪数据的理解，我们通过大量的实验对这些配置进行了深入的探索，得出的主要的实验结论是，用基于反向翻译的方法生成的伪数据来预训练模型是最有效的。我们的实验在多个基准测试集 (CoNLL-2014 测试集和 BEA-2019 共享任务官方测试集) 上实现了最先进性能，并且无需对模型架构进行任何修改。我们还根据语法错误类型和文本的语言水平对模型进行了深入分析。最后，我们提出了未来进一步提高模型性能的方向。

关键字： 自然语言处理；语言生成；语法重写系统；机器翻译

第 1 章　介绍

许多研究将语法纠错（GEC）看成一种机器翻译任务，将不符合语法的句子作为源语言，将符合语法的句子作为目标语言。这种方法适用于最先进的神经机器翻译模型。例如，最初被提出的用于机器翻译的编码器-解码器（EncDec）模型[1]-[4]，已广泛应用于 GEC，并且效果显著[5]-[13]。

然而，将 EncDec 应用于 GEC 的一个挑战是 EncDec 需要大量的训练数据[14]。然而，GEC 最大的公开并行数据集（Lang-8）仅包含 200 万个句子对[15]。可用的数据量不足以支撑模型归纳出的各种语法错误。因此，通过合并伪训练数据[9]-[13],[16] 来增强数据的方法引起了学术界的广泛研究。

在合并伪数据时，必须确定以下几个实验配置，(i) 生成伪数据的方法，(ii) 伪数据的种子语料库，以及 (iii) 优化设置。然而，在 GEC 领域尚未就这些决定达成共识。例如，[9] 发现反向翻译[17] 方法的一个变体（BACKTRANS(NOISY)）优于从原始句子生成伪数据（DIRECTNOISE）。而目前最先进的模型[12] 和 BEA-2019 共享任务的获胜者[13],[18]，都使用了基于 DIRECTNOISE 的方法。

在本研究中，我们研究了上述决策，目的是为学术界提供针对融合伪数据的更好的理解。通过大量的实验，我们探索并确定了合适的 GEC 设置。此外，我们通过评估其在基准数据集上的性能来验证该设置的可靠性。具体来说，在没有任何特定任务的技术或架构的情况下，我们的 EncDec 方法不仅优于以前所有的单模型结果，还优于所有集成结果，除了 Grundkiewicz 等人[13] 的集成结果。通过应用额外的任务特定技术，我们进一步提高了模型性能，并在 CoNLL-2014 测试集和 BEA-2019 共享任务测试集上取得了最先进的性能。

值得注意的是，该研究的初步版本是在一个国际会议（EMNLP-IJCNLP 2019[19]）上提出的。我们对会议论文的扩展主要包括一个附加的实验（第 4 章第 1 节），对实验的深入分析（第 5 章）以及对之前关于 GEC 伪数据研究的全面概述（第 6 章）。具体来说，我们做了以下几点补充:

(i) 我们使用 Grundkiewicz 等人[13] 提出的附加伪数据生成方法进行了实验（第 4 章第 2 节，表 3）。

(ii) 我们分析了不同语言下改变种子语料库对模型性能的影响（第 5 章第 1 节，图 5、图 6）。

(iii) 我们通过比较每种语法错误类型的基线、PRETLARGE 和 PRETLARGE+SSE 模型的性能来分析融合伪数据的有效性（第 5 章第 2 节，图 9）。

(iv) 我们分析了不同语言水平下伪数据的性能可伸缩性（第 5 章第 2 节，图 9）。

(v) 我们总结了前人的研究，提出了生成伪数据的方法，包括基于规则/概率的方法和基于机器翻译的方法（第 6 章第 1 节）。

(vi) 我们证实了以前的研究使用了不同类型的种子语料库（第 6 章第 2 节）。

(vii) 我们总结了以前的研究中使用伪数据进行模型训练的情况（第 6 章第 3 节）。

此外，我们还作了若干修订:

(i) 我们通过公开预训练模型和模型输出结果来确保实验的可重复性。

(ii) 我们的会议论文[19] 使用了 BEA-valid 的随机下采样版本作为验证集。而在本研究中，我们使用了整个 BEA-valid 评估模型，并且确保关于 GEC 实验设置的结论保持不变。

(iii) 部分实验使用 Gigaword 而不是 Wikipedia 作为种子语料库，因为我们确定 Gigaword 是最好的种子语料库（表 3）。

(iv) 为了更简洁地描述 GEC 任务，我们在第二章中添加了若干个数学公式。

第 2 章　问题公式和符号

在这一章中，我们正式定义了本文所涉及的 GEC 任务。设 D 为 GEC 的训练数据，由一个不合语法的源句 X 和一个符合语法的目标句 Y 组成，即 $D = (X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n)$。$|D|$ 表示数据集 D 中句子对的数量。

EncDec 是目前 GEC 任务[6]-[8] 使用的主要方法。为了描述 EncDec，我们定义 X 由词序列 I 组成，即 $X = (x_1, ..., x_I)$，x_i 表示序列 X 的第 i 个词元；定义 Y 总是包含两个额外的特殊标记，$< bos >$ 为 y_0，$< eos >$ 为 y_{J+1}。因此，$Y = (y_0, y_1, ..., y_J, y_{J+1})$，即 Y 的长度是 $J + 2$。所以 EncDec 对以下条件概率进行建模:

$$p(Y|X) = \prod_{j=1}^{J+1} p(y_j|y_{0:j+1}, X, \Theta) \tag{1}$$

Θ 表示模型中所有可训练的参数。我们的目标是为给定的训练数据 D 找到使目标函数 $L(D,\Theta)$ 最小化的最优参数集 $\hat{\Theta}$:

$$L(D,\Theta) = -\frac{1}{|D|}\sum_{(X,Y)\in D}\log(p(Y|X,\Theta)) \tag{2}$$

其中 $p(Y|X,\Theta)$ 表示 Y 关于 X 的条件概率。在标准的监督学习设置中，并行数据集 D 只包含“真正的”并行数据 D_g。然而，在 GEC 中，通常会合并由语法句子 $Y \in \Gamma$ 生成的伪数据 D_p，其中 Γ 表示种子语料库[9],[12],[13]。

我们的兴趣在于以下三个重要方面。(i) 生成伪数据 D_p 有多种方法（第三章）。(ii) 种子语料库 Γ 的选项很多。据我们所知，种子语料库域对模型性能的影响还有待证明。我们比较了三个语料库，即 Wikipedia，SimpleWiki 和 English Gigaword 作为第一次尝试。Wikipedia 和 SimpleWiki 有相似的域，但语法复杂程度不同。因此，可以通过比较这两个语料库来研究语法复杂性是如何影响模型性能的。假设在三个语料库中，Gigaword 包含的噪声最少。所以我们可以使用 Gigaword 来研究纯文本是否可以提高模型性能。(iii) 将 D_p 纳入优化有两个主要方法: 第一种为联合使用两个数据集，并将其合并为 $D = D_g \cup D_p$，然后求解如下的最小化问题:

$$\hat{\Theta} = \underset{\Theta}{argmin}\left\{L\left(D_g \cup D_p, \Theta\right)\right\} \tag{3}$$

我们在下文称此优化方法为 JOINT。另一种方法是使用 D_p 进行预训练，即最小化 $L(D_g,\Theta)$ 后得到 Θ'，然后通过最小化 $L(D_g,\Theta')$ 对模型进行微调。具体优化操作如下:

$$\Theta' = \underset{\Theta}{argmin}\left\{L\left(D_p, \Theta\right)\right\} \tag{4}$$

$$\hat{\Theta} = \underset{\Theta}{argmin}\left\{L\left(D_g, \Theta'\right)\right\} \tag{5}$$

我们将这种优化方法称为 PRETRAIN。

第 3 章　生成伪数据的方法

在本章中，我们将描述生成伪数据的三种方法：反向翻译噪声（BACKTRANS(NOISY)）、直接噪声（DIRECTNOISE）及其变体（DIRECTNOISE(SPELL)）。在第四章中，我们通过实验比较了这些方法。每种生成方法的示例如图 1所示。

Original:	He died there , but the death date is not clear .
BACKTRANS (NOISY):	He died at there , but death date is not clear .
DIRECTNOISE:	⟨mask⟩ ⟨mask⟩ ⟨mask⟩ , 2 but ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ is not ⟨mask⟩ ⟨mask⟩
DIRECTNOISE (SPELL):	H@@ o died there , but the death te is not clear .
Original:	Gre@@ en@@ space Information for G@@ rea@@ ter London .
BACKTRANS (NOISY):	The information for Gre@@ en@@ space information about G@@ rea@@ ter London .
DIRECTNOISE:	⟨mask⟩ ⟨mask⟩ ⟨mask⟩ for ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ ⟨mask⟩
DIRECTNOISE (SPELL):	Gre@@ en@@ space Information cp@@ r@@ ts for G@@ rea@@ ter London .
Original:	The cli@@ p is mixed with images of Toronto streets during power failure .
BACKTRANS (NOISY):	The cli@@ p is mix with images of Toronto streets during power failure .
DIRECTNOISE:	The ⟨mask⟩ is mixed ⟨mask⟩ images si@@ of The ⟨mask⟩ streets large ⟨mask⟩ power R@@ failure place ⟨mask⟩
DIRECTNOISE (SPELL):	The V@@ li@@ p is mixed with images of Toronto streets during power failure .
Original:	At the in@@ stitute , she introduced tis@@ sue culture methods that she had learned in the U.@@ S.
BACKTRANS (NOISY):	At in@@ stitute , She introduced tis@@ sue culture method that she learned in U.@@ S.
DIRECTNOISE:	⟨mask⟩ the the ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ ⟨mask⟩ tis@@ culture R@@ methods , she P ⟨mask⟩ the s U.@@ ⟨mask⟩
DIRECTNOISE (SPELL):	At the in@@ stitute , she introduced tis@@ sue culture methods that she had l@@ Gen@@ em@@ d the U.@@ S.

图 1: 不同方法生成的句子示例

3.1　反向翻译噪声（BACKTRANS(NOISY)）

EncDec 模型的反向翻译方法最初是由 Sennrich 等人[17] 提出的。反向翻译方法训练了一个反向模型，从给定的目标（语法）句子 Y 生成一个源（非语法）句子 X，然后将反向模型的输出与输入配对，作为伪数据使用。

目前，在机器翻译研究领域，由于反向翻译具有较强的性能[20],[21]，被认为是生成伪数据的标准方法。然而，对于 GEC 来说并非如此。Xie 等人[9] 报告说，单纯地应用反向翻译只会导致性能的微小改善，因为在原始的反向翻译中，模型通常过于保守，无法产生足够数量的语法错误。为了克服这个问题，Xie 等人[9] 提出了一种反向翻译的变体，称为 BACKTRANS。该方法将 $r\beta_{random}$ 随机添加到每一步的每个假设的分数中。这里，标量噪声 r 从区间 [0,1] 中均匀采样，$\beta_{random} \in R_{\geq 0}$ 是控制噪声尺度的超参数。如果我们设置 $\beta_{random} = 0$，那么 BACKTRANS (NOISE) 与原始反向翻译相同。

3.2 直接噪声:DIRECTNOISE 和 DIRECTNOISE(SPELL)

BACKTRANS (NOISE) 使用反向模型生成不符合语法的句子，而 DIRECTNOISE 将噪声“直接”加入到符合语法的句子中[12],[13],[16],[21]。具体来说，对于给定句子中的每个词元，该方法概率地选择以下操作之一：(i) 使用掩码，(ii) 删除，(iii) 插入从正态分布中采样的随机词元，以及 (iv) 保留原始词元。具体算法在算法 1 中给出。对于每个词元，选择都是基于分类分布。

Algorithm 1: DIRECTNOISE Algorithm

Data: Grammatical sentence $\boldsymbol{Y} \in \mathcal{T}$
Result: Pseudo Corpus $\mathcal{D}_p$

$\mathcal{D}_p = \{\}$ // create empty set
$\boldsymbol{\mu} = (\mu_{\text{mask}}, \mu_{\text{deletion}}, \mu_{\text{insertion}}, \mu_{\text{keep}})$ s.t. $\Sigma\boldsymbol{\mu} = 1$
for $\boldsymbol{Y} \in \mathcal{T}$ **do**
  $\boldsymbol{X} = (\ )$
  for $j \in (1, \ldots, J)$ **do**
    $action \sim Cat(action|\boldsymbol{\mu})$
    if *action is* keep **then**
      append y_j to $\boldsymbol{X}$
    else if *action is* mask **then**
      append $\langle mask \rangle$ to $\boldsymbol{X}$
    else if *action is* deletion **then**
      continue
    else if *action is* insertion **then**
      append y_j to $\boldsymbol{X}$
      $w \sim unigram_distribution(\mathcal{D}_g)$
      append w to $\boldsymbol{X}$
  $\mathcal{D}_p = \mathcal{D}_p \cup \{(\boldsymbol{X}, \boldsymbol{Y})\}$

最近，Grundkiewicz 等人[13] 提出了另一种生成伪数据的方法，其核心思想是使用现成的拼写检查器为给定的单词生成混淆集，然后概率地用混淆集中的单词替换。这种方法一般可以理解为 DIRECTNOISE 的一种变体，它放弃了掩蔽操作，采用了替换操作。因此，我们将该方法称为 DIRECTNOISE (SPELL)。

第 4 章　实验

我们实验的目标是研究第二章中介绍的三个方面，并设计实验来确保实验结果是可重复的，适用于大部分 GEC[22]。具体来说，实验基于以下两种策略。(i) 使用现成的 EncDec 模型，没有任何特定任务的架构或技术。(ii) 对验证集上的每个方法和设置执行超参数调优、评估和比较。

4.1　实验设置

数据集　BEA-2019 官方数据集[18] 是我们实验的训练、验证和测试数据的来源。该数据集由以下语料库组成: 剑桥第一证书语料库[23]，Lang-8 学习英语语料库[15],[24]，新加坡国立大学学习英语语料库（NUCLE）[25] 和 W&I+LOCNESS[26],[27]。我们将训练数据称为 BEA-train。

模型　每个实验采用 Transformer EncDec 模型[4]。具体来说，我们使用 fairseq 工具包[36] 和 Vaswani 等人[4] 的 “Transformer(big)” 设置，其中编码器和解码器都有六层，每层有 16 个注意力头，单词嵌入大小 d_{model} 为 1024，前馈网络大小 d_{ff} 为 4096，丢失率设置为 0.3。

优化　我们比较了两种优化方法，即 JOINT 和 PRETRAIN。JOINT 方法使用 Adam[37] 对模型进行优化。而 PRETRAIN 方法则使用 Adam 预训练模型，并用 Adafactor[38] 在 BEA-train 上对其进行微调。每种设置的详细超参数如表 1 和表 2 所示。

表 1：JOINT 优化的超参数

Model Architecture	Transformer [4] (“big” setting)
Optimizer	Adam [37] ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. [4]
Number of Epochs	40
Dropout	0.3
Stopping Criterion	Train model for 40 epochs. During the training, save model parameter for every epoch. Then take average of last 20 checkpoints.
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Beam Search	Beam size 5 with length-normalization

表 2：预训练优化的超参数

Pretraining	
Model Architecture	Transformer [4] ("big" setting)
Optimizer	Adam [37] ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Same as described in Section 5.3 of Vaswani et al. [4]
Number of Epochs	10
Dropout	0.3
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Fine-tuning	
Model Architecture	Transformer [4] ("big" setting)
Optimizer	Adafactor [38]
Learning Rate Schedule	Constant learning rate of 3×10^{-5}
Number of Epochs	30
Dropout	0.3
Stopping Criterion	Use the model with the best validation perplexity on BEA-valid
Gradient Clipping	1.0
Loss Function	Label smoothed cross entropy (smoothing value: $\epsilon_{ls} = 0.1$) [39]
Beam Search	Beam size 5 with length-normalization

4.2 伪数据生成

我们比较了生成伪数据的 BACKTRANS(NOISE)、DIRECTNOISE 和 DIRECTNOISE(SPELL) 方法的有效性。为此，我们首先研究适用于 BACKTRANS(NOISE) 和 DIRECTNOISE 的超参数，然后比较 BACKTRANS(NOISE)、DIRECTNOISE 和 DIRECTNOISE(SPELL)。值得注意的是，在本节中，我们使用 (i)JOINT 方法和 (ii) 所有 SimpleWiki 作为种子语料库 Γ。

如第三章所述，DIRECTNOISE 包含四个超参数 (μ_{mask}, $\mu_{deletion}$, μ_{insert}, μ_{keep})。对所有这些参数进行网格搜索的计算成本很高。因此，本研究中只关注 μ_{mask} 的影响。我们故意设置 $\mu_{keep} = 0.2$，并使用 $\mu_{deletion} = \mu_{insert} = (1 - \mu_{keep} - \mu_{mask})/2$。结果如图 2 所示。其中 $\mu_{mask} = 0.3$ 性能最好。因此，我们在剩下的实验中使用 $\mu_{mask} = 0.3$。

我们还通过评估 BACKTRANS(NOISE) 在 BEA-valid 上的性能来研究 β_{random} 的变化对 BACKTRANS(NOISE) 的影响。通过实验发现，$\beta_{random} = 6$ 的性能最好。值得注意的是，根据图 3，$\beta_{random} = 0$ 的性能弱于基线，这是因为在没有噪声的情况

下，反向模型会变得过于保守，不会产生语法错误，这与 Xie 等人[9] 报道的现象是一致的。因此，伪数据不能为模型提供有用的教学信号并最终影响模型的性能。

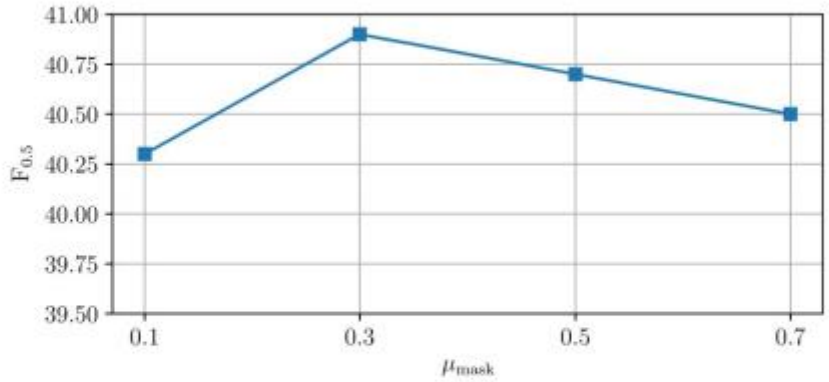


图 2: 不同参数 DIRECTNOISE(μ_{mask}) 下，模型在 BEA-valid 上的性能

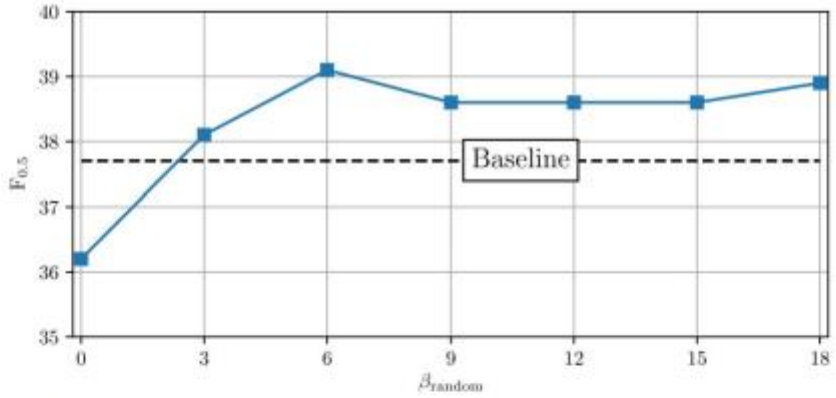


图 3: 不同参数 BACKTRANS(NOISE)(β_{random}) 下，模型在 BEA-valid 上的性能

第 5 章　分析

在第 4 章中，我们证实了 PRETLARGE 和 PRETLARGE+SSE 配置实现了优于当前顶级模型的性能。在本章中，我们提出了未来进一步提高性能的方向，并通过分析语法错误类型表现和文本语言水平的实验结果来实现这一点。具体来说，我们对以下两个方面进行了分析：(i) 伪数据在不同种子语料库之间的有效性如何变化，(ii)PRETLARGE 的优缺点。

5.1 不同种子语料库的有效性

错误类型分析 在本节中，通过 BEA-valid 上的每种错误类型上的表现，我们分析了使用不同种子语料库训练的模型的性能。然而，一个种子语料库是否比另一个语料库更有效地纠正某些语法错误类型。为此，我们使用表 3 所示的 DIRECTNOISE 模型。图 4 给出了结果。从图中可以看出不同的种子语料库确实具有不同的特征。例如，Gigaword 在错误类型 (如 PUNCT、PREP 和 SPELL) 上优于其他种子语料库。另一方面，SimpleWiki 在 VERB:TENSE 错误方面优于 Gigaword。

表 3：模型在 BEA-valid 上的表现

Method	Seed Corpus $\mathcal{T}$	Prec.	Rec.	$F_{0.5}$
Baseline	N/A	45.2	22.7	37.7
BACKTRANS (NOISY)	Wikipedia	42.8	30.5	39.6
BACKTRANS (NOISY)	SimpleWiki	41.7	31.2	39.1
BACKTRANS (NOISY)	Gigaword	42.2	33.1	40.0
DIRECTNOISE	Wikipedia	47.1	25.8	40.4
DIRECTNOISE	SimpleWiki	48.3	25.4	40.9
DIRECTNOISE	Gigaword	47.3	26.7	41.0

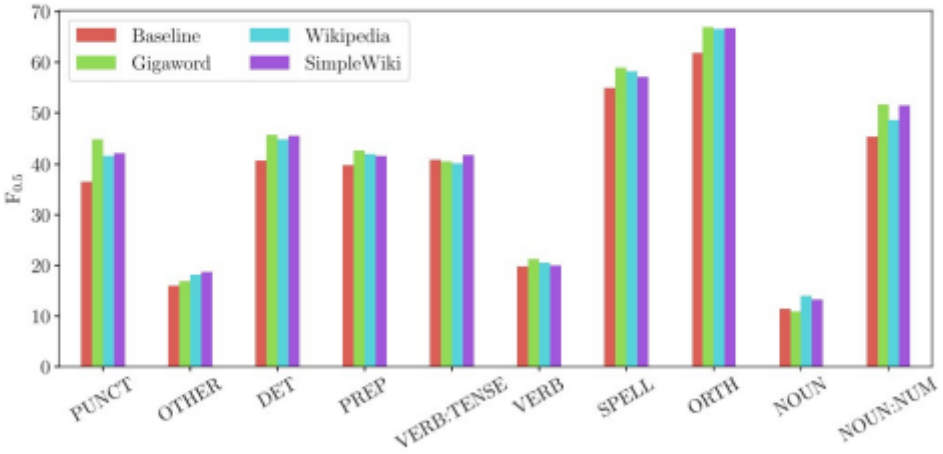


图 4: 不同种子语料库在不同错误类型上的 BEA-valid 性能。这里只列出了 10 种最常见的错误类型。

语言水平分析 BEA-valid 的一个显著特点是由 A、B、C 和 N 四个部分组成，每个部分的英语水平不同。这里，A（初级）、B（中级）和 C（高级）是从 CEFR 级别[50]衍生出来的。剩下的级别（N）对应于以英语为母语的人所写的文本。这些英文水平为我们提供了对模型更深入的了解。如图 5 所示，不同英文水平的误差分布有显著差异。例如，DET 错误在水平 A、B 和 C 中很常见，但在 N 中不常见。

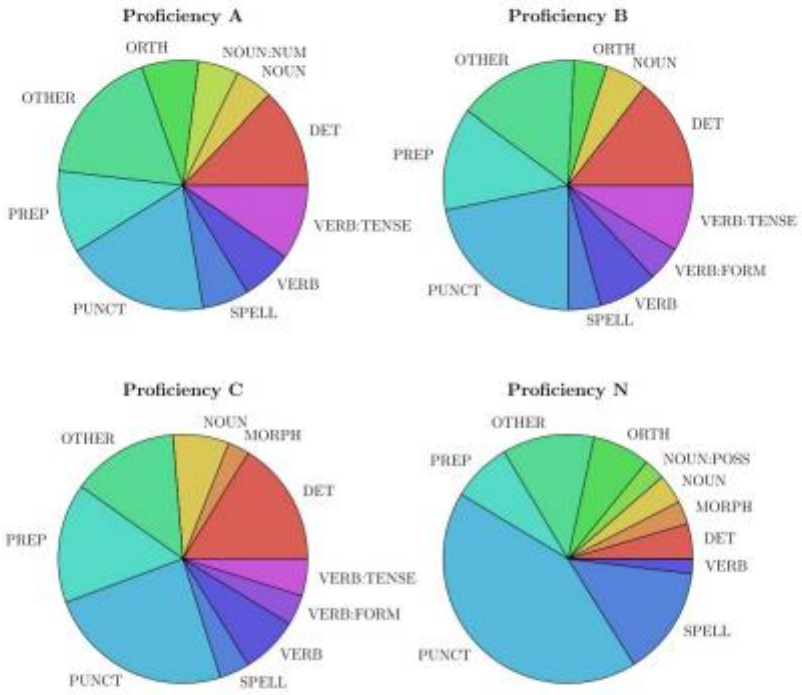


图 5: 不同语言水平的错误类型分布。这里只列出了 10 种最常见的错误类型。

我们感兴趣的是对于每种语言水平更改种子语料库对模型性能的影响，因为每种语言水平需要不同的种子语料库。例如，SimpleWiki 适合水平 A、B 和 C，因为它的语法复杂性更接近英语学习者。类似地，Gigaword 适用于水平 N，因为它的文本是由母语为英语的人编写的。我们使用表 3 所示的 DIRECTNOISE 模型分析了种子语料库和每种语言水平之间的关系，具体来说，我们评估了每种语言水平模型的性能。结果如图 6 所示，该图表明在语言水平 A、B 和 C 上，SimpleWiki 的性能与 Gigaword 相当，甚至更好。然而，Gigaword 在语言水平 N 上优于其他两个语料库。这些结果支持了我们的假设，即种子语料库的语法复杂性与语言水平之间存在一定的关系。

一个特定的种子语料库比其他语料库更适合特定的语言水平，这与我们在第 5

章第 1 节中的发现是一致的。例如，根据图 6，动词时态错误在语言水平 A、B 和 C 的 10 个最常见的错误中，在 5.1 节中，我们发现 SimpleWiki 在纠正动词时态错误方面表现最好。这可能是图 6 中 SimpleWiki 在水平 A、B 和 C 上表现出色的原因之一。

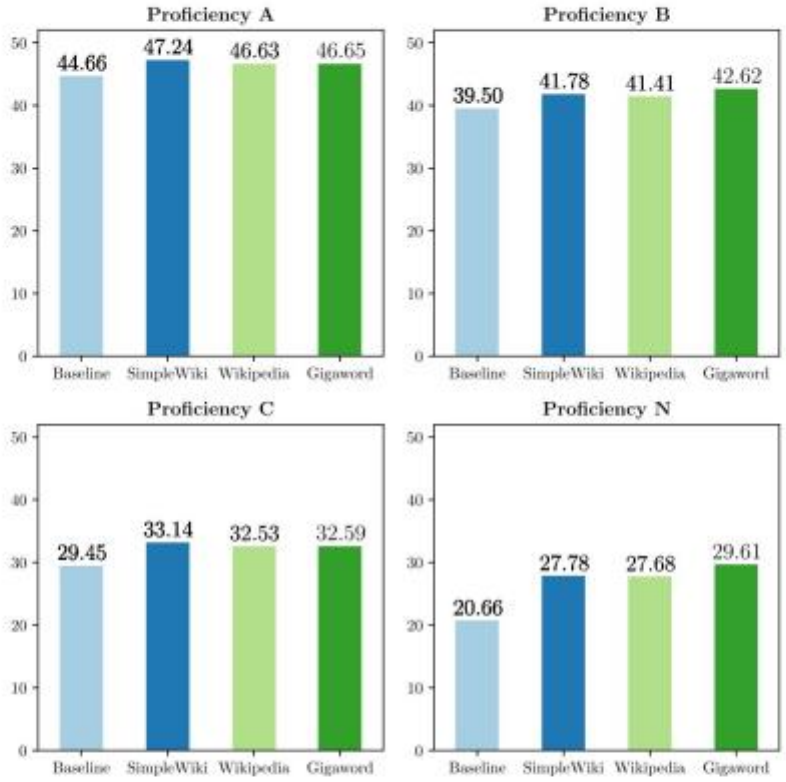


图 6: 种子语料库对语言水平 A、B、C、N 的影响。

5.2 PRETLARGE 的优缺点

错误类型分析 我们通过每种错误类型在 BEA-valid 上的表现来分析每个模型得性能（图 8）。具体来说，我们感兴趣的是与基线模型相比，PRETLARGE 和 PRETLARGE+SSE 的性能，基线模型仅使用真实数据 D_g 进行训练。值得注意的是，BEA-valid 为每个编辑保留错误类型注释，由 ERRANT 自动注释。

图 8 显示，与基线模型相比，PRETLARGE 提高了所有错误类型的性能。此外，令人惊讶的是，PRETLARGE+SSE 不仅在拼写错误类型上提高了 PRETLARGE 的性能，而且在大多数其他错误类型上也提高了性能。因此我们推测，加入 SSE 可以使模型更加健壮。

图 8 也表明了我们的模型的一个主要弱点，即它们在实义词错误（如名词和动词）方面的表现相对较差。这个缺点在 GEC 模型中普遍存在。在参与 BEA-2019 共享任务[18] 的系统错误类型性能中也观察到类似的趋势。原因之一是真实数据和伪数据中的内容词错误数量都不够。例如，如图 5 所示，NOUN 和 VERB 错误的比例小于 PREP、PUNCT 和 DET 等错误的比例。因此，开发一种专门生成实义词错误的方法是未来工作的重要方向。

对图 7 中的一个例子进行定性分析为我们改进 GEC 模型提供了其他方向。在这里，模型通过插入缺失的“the”词元成功地纠正了 DET 错误。但是，模型忽略了句子中的两个名词错误: 一个是将“cloakroom”替换为“dressing room”，另一个是将“signature”替换为“autograph”。对于前者，模型需要知道“musicians”应该在“dressing room”而不是在“cloakroom”。一种可能的方法是开发一种将外部常识注入模型的方法。对于后一种错误，源句子没有包含足够的信息让模型进行修正。这是因为源句中的“we”不能消除歧义：如果“we”是观众，“signature”应该更正为“autograph”。然而，假设“we”指的是唱片公司的人，“signature”似乎是合适的。开发跨句 GEC 模型[41] 是克服这种歧义困难的一种很有前途的方法。

Source Sentence:	When the concert finished , we went to cloakroom to get signatures from musicians .
Gold Sentence:	When the concert finished , we went to **the** dressing room to get autographs from musicians .
Model Output:	When the concert finished , we went to **the** cloakroom to get signatures from musicians .

图 7: 由我们的模型产生的名词错误。粗体文本表示模型成功纠正的语法错误，带下划线的文本表示模型未纠正的语法错误。

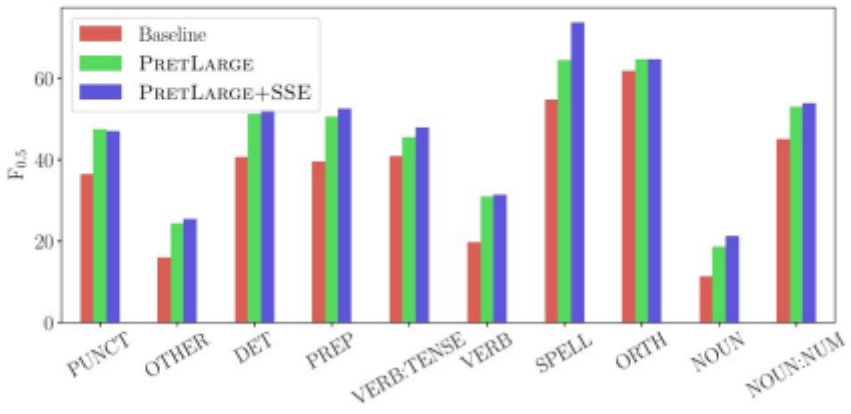


图 8: 模型在各种错误类型的 BEA-valid 上的性能

语言水平分析 我们研究了增加伪数据量对每种语言水平的影响。如果达到某种语言水平的性能饱和，那么除了增加伪数据量之外，还需要其他方法来提高性能。结果如图 9 所示，语言水平等级 A 和 N 的表现与伪数据量成比例，而 B 和 C 似乎达到了饱和。正如第 5 章第 1 节所讨论的，不同的种子语料库适用于不同的语言水平。因此，我们可以通过合并一个语法复杂度较低的种子语料库来提高 B 和 C 的性能。然而，SimpleWiki 的大小仅包含大约 1.4M 个句子，不足以达到这个目的。因此，从原始语料库（例如 Common Crawl16）中提取文本是至关重要的。

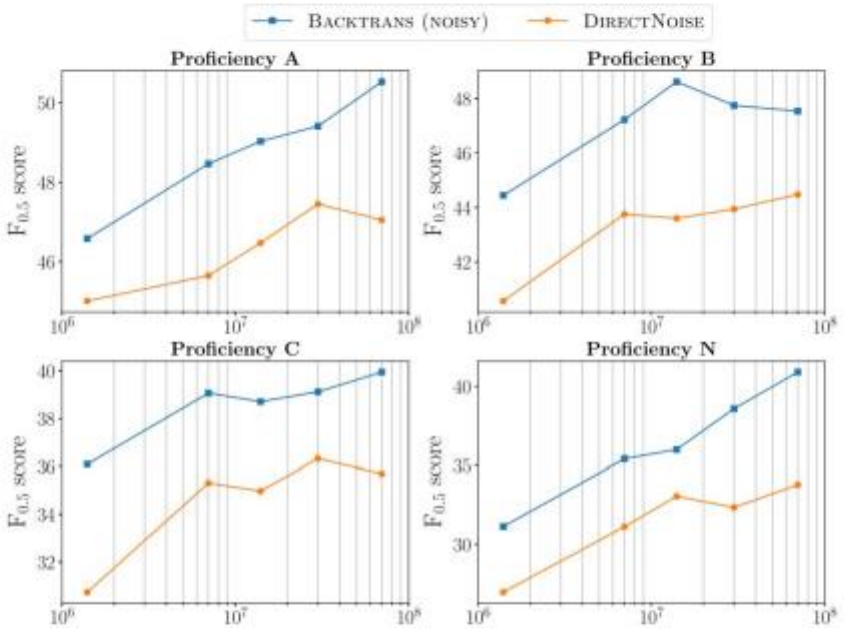


图 9: 模型在 BEA-valid 的每种语言水平上的表现

第 6 章 相关工作

6.1 生产伪数据的方法

缺乏真实的数据 D_g 一直是 GEC 领域的一个挑战。缓解这个问题的核心方法是伪数据的生成。一般有两种方法，基于规则/概率的方法和基于机器翻译的方法。

一种常见的基于规则/概率的方法将错误模板应用于外部语法句子（即种子语料库），而模板是由少量真实数据生成的。这种方法的优点之一是可以很容易地操纵模板。例如，它可以被设计成专注于特定的错误类型，如名词[51]，冠词[52]-[54] 和介

词[54]-[56]，或者它可以支持一般[16],[57],[58]的语法错误。最近，Grundkiewicz 等人[13]提出了一个类似的方法。他们用现成的英语拼写检查器创建的混淆集来替换单词，并且在 BEA2019 共享任务[18]中取得了最好的表现。

另一种基于规则/概率的方法是赵等人[12]提出的方法，即在语法句子中注入合成噪声。该方法在很大程度上受到去噪自动编码器[59]和大型预训练语言模型[12],[13],[42],[60]的启发。我们使用这种方法的变体（DIRECTNOISE）进行了一个实验。

基于规则/概率的方法的一个主要缺点是，很难获得与真实数据中存在的错误（如自然发生的语法错误）非常相似的高质量数据，一些研究报道了伪数据可能会导致性能下降的情况[57],[58],[61]。最近，利用模型从语法句子中产生错误的机器翻译方法受到了研究界的关注，模型通常是统计机器翻译[62]或神经机器翻译[9],[11],[63]。例如，Lichtarget 等人[11]提出了一种双向翻译。该方法将机器翻译模型产生的翻译错误视为伪语法错误。首先将给定的语法句子翻译成任意的过度语言（例如日语），然后将句子翻译回原语言，作为伪数据的源句。另一种基于机器翻译的方法是 Xie 等人[9]提出的反向翻译方法的变体，即 BACKTRANS（NOISE）。他们证明了 BACKTRANS（NOISE）可以生成与真实文本相似的句子。在实验中，我们将 BACKTRANS（NOISE）与 DIRECTNOISE 进行了比较，结果表明前者具有更好的性能。

除了生成伪数据之外，还存在一种从 Web 抓取伪数据的方法。具体来说，lichtarget 等人[11]提取维基百科的修订历史构造伪数据。这种方法与我们的研究是相关的，可以联合使用生成的伪数据和维基百科修订版。

6.2 种子语料库

历史上，许多类型的语料库被认为是生成伪数据的种子语料库，如新闻文章集合[9],[12],[13],[51],[55]，英国国家语料库[61]，英语维基百科[53],[55],[56],[58]。这一事实表明，在种子语料库的选择方面尚未达成共识。我们关于种子语料库对模型性能的有效性的研究问题正是基于这样的情况而开展的。

Felice 和 Yuan[58]的研究与我们的相似。作者指出，种子语料库的变量包括主题、体裁、风格、文本复杂度和作者的母语。但是，他们只把英文维基百科作为种子语料库，对于这些变量不能得到充分的结论。在本研究中，我们进行了对照实验，并比较了三种不同的种子语料库。我们发现 Gigaword 是最好的种子语料库，至少在我们的实验中是这样。

6.3 优化器设置

如第二章所述，至少有两种优化方法，即 JOINT 和 PRETRAIN。在非神经模型[57],[58] 和神经模型[9] 中都观察到了 JOINT 优化的有效性。然而，GEC 领域正在积极探索 PRETRAIN 方法。

关于 GEC 模型的 PRETRAIN 方法的起源是 EncDec 的“部分预训练”。Junczys-Dowmunt 等人[8] 分别用 Word2Vec 向量[64] 和预训练的语言模型初始化 EncDec 的嵌入矩阵和解码器参数。此外，他们报告说这两种方法都能持续提高 EncDec 的性能。Chollampatt 和 Ng[41] 也使用了类似的方法。然而，由于 EncDec 的“全预训练”具有较强的经验性能，已经成为[11]-[13],[16] 的主导。该方法使用伪数据预训练整个 EncDec，然后使用真实数据对其进行微调。

GEC 的一个有趣的方面是，尽管它与机器翻译合并了相同的模型（EncDec），但主要的优化方法是不同的。JOINT 通常用于机器翻译[21],[25]，而 PRETRAIN 用于现有的 GEC 研究。我们比较了这两种优化方法，并确认了 PRETRAIN 的优越性。

第 7 章 总结

在本研究中，我们研究了 GEC 中伪数据合并的几个方面。通过大量的实验，我们总结出了以下步骤，可以有效地训练模型并获得较强的性能:

(i) 使用 Gigaword 作为种子语料库。

(ii) 用大量 BACKTRANS(NOISE) 数据对模型进行预训练。

(iii) 使用真实数据对预训练模型进行微调。

此外，我们通过使用 CoNLL-2014 和 BEA-2019 测试集实现最先进的性能，证明了这一建议的有效性。

随后，我们对实验结果进行了深入分析; 我们的发现可以总结如下:

(i) 合适的种子语料库因不同的语法错误类型和熟练程度而异。例如，低语法复杂性的种子语料库适合低语言水平的文本。

(ii) 与基线相比，我们提出的设置 (PRETLARGE) 提高了所有语法错误类型的性能。然而，实义词错误仍然是一个挑战。

(iii) 就伪数据量而言，每种语言水平的性能都显示出有希望的可伸缩性。然而，当使用最大的数据时，语言水平 B 和 C 的表现似乎饱和了。

基于这些观察，可以提出几种可能的方法来进一步提高模型的性能，特别是在实义词错误方面。例如，(1) 开发一种可以专门生成实义词错误的伪数据生成方法，(2) 将常识知识注入模型，以及 (3) 更复杂的跨句 GEC 是有前途的方法。

参考文献

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Proc. Adv. Neural Inf. Process. Syst. 28, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in Proc. 3rd Int. Conf. Learn. Representations, 2015.

[3] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention based neural machine translation," in Proc.Conf.Empirical Methods Natural Lang. Process., 2015, pp. 1412–1421.

[4] A. Vaswani et al., "Attention is all you need," in Proc. Adv. Neural Inf. Process. Syst. 31, 2017, pp. 5998–6008.

[5] Z. Yuan and T. Briscoe, "Grammatical error correction using neural machine translation," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2016, pp. 380–386.

[6] J. Ji, Q. Wang, K. Toutanova, Y. Gong, S. Truong, and J. Gao, "A nested attention neural hybrid model for grammatical error correction," in Proc. 55th Annu. Meeting Assoc. Comput. Linguist., 2017, pp. 753–762.

[7] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction,"in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 5755–5762.

[8] M. Junczys-Dowmunt, R. Grundkiewicz, S. Guha, and K. Heafield, "Approaching neural grammatical error correction as a low-resource machine translation task," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2018, pp. 595–606.

[9] Z. Xie, G. Genthial, S. Xie, A. Ng, and D. Jurafsky, "Noising and denoising natural language: Diverse backtranslation for grammar correction," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2018, pp. 619–628.

[10] T. Ge, F. Wei, and M. Zhou, "Fluency boost learning and inference for neural grammatical error correction," in Proc. 56th Annu. Meeting Assoc. Comput. Linguist., 2018, pp. 1055–1065.

[11] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. Tong, "Corpora generation for grammatical error correction," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2019, pp. 3291–3301.

[12] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, "Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2019, pp. 156–165.

[13] R. Grundkiewicz, M. Junczys-Dowmunt, and K. Heafield, "Neural grammatical error correction systems with unsupervised pre-training on synthetic data," in Proc. 14th Workshop Innovative Use NLP Building Educ. Appl., 2019, pp. 252–263.

[14] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in Proc. 1st Workshop Neural Mach. Transl., 2017, pp. 28–39.

[15] T. Mizumoto, M. Komachi, M. Nagata, and Y. Matsumoto, "Mining revision log of language learning SNS for automated japanese error correction of second language learners," in Proc. 5th Int. Joint Conf. Natural Lang. Process., 2011, pp. 147–155.

[16] Y. J. Choe, J. Ham, K. Park, and Y. Yoon, "A neural grammatical error correction system built on better pre-training and sequential transfer learning," in Proc. 14th Workshop Innovative Use NLP Building Educa. Appl., 2019, pp. 213–227.

[17] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in Proc. 54th Annu. Meeting Assoc. Comput. Linguist., 2016, pp. 86–96.

[18] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 shared task on grammatical error correction," in Proc. 14th Workshop Innovative Use NLP Building Educa. Appl., 2019, pp. 52–75.

[19] S. Kiyono, J. Suzuki, M. Mita, T. Mizumoto, and K. Inui, "An empirical study of incorporating pseudo data into grammatical error correction," in Proc. Conf. Empirical Methods Natural Lang. Process. and 9th Int. Joint Conf. Natural Lang. Process., 2019, pp. 1236–1242.

[20] B. Haddow et al., "The University of Edinburgh' s submissions to the WMT18 news translation task," in Proc. 3rd Conf. Mach. Transl., 2018, pp. 399–409.

[21] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding backtranslation at scale," in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 489–500.

[22] X. Bouthillier, C. Laurent, and P. Vincent, "Unreproducible research is reproducible," in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 725–734.

[23] H. Yannakoudakis, T. Briscoe, and B.Medlock, "A new dataset and method for automatically grading ESOL texts," in Proc. 49th Annu. Meeting Assoc. Comput. Linguist., 2011, pp. 180–189.

[24] T. Tajiri, M. Komachi, and Y. Matsumoto, "Tense and aspect error correction for ESL learners using global context," in Proc. 50th Annu. Meeting Assoc. Comput. Linguist., 2012, pp. 198–202.

[25] D. Dahlmeier, H. T. Ng, and S. M. Wu, "Building a large annotated corpus of learner English: The NUS corpus of learner english," in Proc. 8th Workshop Building Educa. Appl. Using NLP, 2013, pp. 22–31.

[26] H. Yannakoudakis, Ø. E. Andersen, A. Geranpayeh, T. Briscoe, and D. Nicholls, "Developing an automated writing placement system for ESL learners," Appl. Meas. Edu., vol. 31, no. 3, pp. 251–267, 2018.

[27] S. Granger, "The computer learner corpus: A versatile new source of data for SLA research," in Learner English on Computer, S. Granger, Ed. London, U.K. and New York, NY, USA: Addison-Wesley, 1998, pp. 3–18.

[28] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units,"in Proc. 54th Annu. Meeting Assoc. Comput. Linguist., 2016, pp. 1715–1725.

[29] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, "The CoNLL-2014 shared task on grammatical error correction," in Proc. 18th Conf. Comput. Natural Lang. Learn.: Shared Task, 2014, pp. 1–14.

[30] C. Napoles, K. Sakaguchi, and J. Tetreault, "JFLEG: A fluency corpus and benchmark for grammatical error correction," in Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist., 2017, pp. 229–234.

[31] C. Bryant, M. Felice, and T. Briscoe, "Automatic annotation and evaluation of error types for grammatical error correction," in Proc. 55th Annu. Meeting Assoc. Comput. Linguist., 2017, pp. 793–805.

[32] M. Felice, C. Bryant, and T. Briscoe, "Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments," in Proc. COLING 26th Int. Conf. Comput. Linguist.: Tech. Papers, 2016, pp. 825–835.

[33] D. Dahlmeier and H. T. Ng, "Better evaluation for grammatical error correction," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2012, pp. 568–572.

[34] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "Ground truth for grammatical error correction metrics," in Proc. 53rd Annu. Meeting Assoc. Comput. Linguist. and 7th Int. Joint Conf. Natural Lang. Process., 2015, pp. 588–593.

[35] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "GLEU without tuning," 2016, arXiv:1605.02592.

[36] M. Ott et al., "fairseq: A fast, extensible toolkit for sequence modeling," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2019, pp. 48–53.

[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learn. Representations, 2015.

[38] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in Proc. 35th Int. Conf. Mach. Learn., 2018, pp. 4603–4611.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. Comput. Vision Pattern Recognit., 2016, pp. 2818–2826.

[40] M. Junczys-Dowmunt and R. Grundkiewicz, "Phrase-based machine translation is state-of-the-art for automatic grammatical error correction," in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 1546–1556.

[41] S. Chollampatt, W. Wang, and H. T. Ng, "Cross-sentence grammatical error correction," in Proc. 57th Annu. Meeting Assoc. Comput. Linguist., 2019, pp. 435–445.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2019, pp. 4171–4186.

[43] R. Grundkiewicz and M. Junczys-Dowmunt, "Near human-level performance in grammatical error correction with hybrid machine translation," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2018, pp. 284–290.

[44] L. Barrault et al., "Findings of the 2019 conference on machine translation (WMT19)," in Proc. 4th Conf. Mach. Transl., 2019, pp. 1–61.

[45] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Agreement on targetbidirectional neural machine translation," in Proc. Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2016, pp. 411–416.

[46] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," in Proc. 1st Conf. Mach. Transl., 2016, pp. 371–376.

[47] R. Sennrich et al., "The University of Edinburgh' s neural MT systems for WMT17," in Proc. 2nd Conf. Mach. Transl., 2017, pp. 389–399.

[48] M. Morishita, J. Suzuki, and M. Nagata, "NTT neural machine translation systems at WAT 2019," in Proc. 6th Workshop Asian Transl., 2019, pp. 99–105.

[49] H. Asano, M. Mita, T. Mizumoto, and J. Suzuki, "The AIP-Tohoku system at the BEA-2019 shared task," in Proc. 14th Workshop Innovative Use NLP Building Educa. Appl., 2019, pp. 176–182.

[50] D. Little, "The common European framework of reference for languages: Content, purpose, origin, reception and impact," Lang. Teach., vol. 39, no. 3, pp. 167–190, 2006.

[51] C. Brockett, W. B. Dolan, and M. Gamon, "Correcting ESL errors using phrasal SMT techniques," in Proc. 21st Int. Conf. Comput. Linguist. and 44th Annu. Meeting Assoc. Comput. Linguist., 2006, pp. 249–256.

[52] E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara, "Automatic error detection in the Japanese Learners' English spoken data," in Proc. Companion Volume Proc. 41st Annu. Meeting Assoc. Comput. Linguist., 2003, pp. 145–148.

[53] A. Rozovskaya and D. Roth, "Training paradigms for correcting errors in grammar and usage," in Proc. Human Lang. Technol.: Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist., 2010, pp. 154–162.

[54] A. Rozovskaya, M. Sammons, and D. Roth, "The UI system in the HOO 2012 shared task on error correction," in Proc. 7th Workshop Building Educa. Appl. Using, 2012, pp. 272–280.

[55] A. Rozovskaya and D. Roth, "Generating confusion sets for context sensitive error correction," in Proc. Conf. Empirical Methods Natural Lang. Process., 2010, pp. 961–970.

[56] A. Cahill, N. Madnani, J. Tetreault, and D. Napolitano, "Robust systems for preposition error correction using wikipedia revisions,"in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol., 2013, pp. 507–517.

[57] Z. Yuan and M. Felice, "Constrained grammatical error correction using statistical machine translation," in Proc. 7th Conf. Comput. Natural Lang. Learn.: Shared Task, 2013, pp. 52–61.

[58] M. Felice and Z. Yuan, "Generating artificial errors for grammatical error correction," inProc. Student Res. Workshop 14th Conf. Eur. Chapter Assoc. Comput. Linguist., 2014, pp. 116–126.

[59] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in Proc. 25th Int. Conf. Mach. Learn., 2008, pp. 1096–1103.

[60] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "MASS: Masked sequence to sequence pre-training for language generation," in Proc. 36th Int. Conf. Mach. Learn., 2019, pp. 5926–5936.

[61] J. Foster and O. Andersen, "GenERRate: Generating errors for use in grammatical error detection," in Proc. 4th Workshop Innovative Use NLP Building Educa. Appl., 2009, pp. 82–90.

[62] M. Rei, M. Felice, Z. Yuan, and T. Briscoe, "Artificial error generation with machine translation and syntactic patterns," in Proc. 12th Workshop Innovative Use NLP Building Educ. Appl., 2017, pp. 287–292.

[63] S. Kasewa, P. Stenetorp, and S. Riedel, "Wronging a right: Generating better errors to improve grammatical error detection," in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 4977–4983.

[64] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in Proc. Adv. Neural Inf. Process. Syst. 27, 2013, pp. 3111–3119.

[65] I.Caswell, C.Chelba, and D.Grangier, "Tagged back-translation," in Proc.4th Conf.Mach.Transl., 2019, pp.53–63.

作者信息



Shun Kiyono 于 2017 年和 2019 年分别从日本东北大学获得工学学士学位和信息科学硕士学位。2019 年加入理化研究所的高级智能项目中心。研究方向包括机器学习和自然语言处理。



Masahiro Kaneko 于 1999 年和 2001 年分别从日本庆应大学获得工程学士学位和工程硕士学位，于 2005 年从日本奈良先端科技大学信息科学研究科获得工程博士学位。2001 年加入了日本电报电话公司的 NTT 通信科学实验室。2018 年加入了日本东北大学信息科学研究科，担任副教授。目前是东北大学数据驱动科学与人工智能中心的教授。研究兴趣包括机器学习和自然语言处理。



Mamoru Komachi 于 2010 年获得甲南大学工学学士学位，于 2012 年和 2015 年分别获得奈良科学技术学院信息科学的工学硕士和博士学位，于 2013 年至 2015 年担任日本学术振兴会特别研究员 (DC2)。2015 年至 2017 年，担任东北大学的博士后研究员。2017 年至 2019 年，担任理化研究所高级智能项目中心的博士后研究员。目前是日本 Future Corporation 的研究工程师，并同时担任理化研究所高级智能项目中心的客座研究员。研究兴趣包括语法纠错、简答评分和自然语言处理。



Mamoru Komachi 是日本东北大学信息科学研究科的教授，也是该校自然语言处理实验室的负责人，同时在理化研究所高级智能项目中心担任自然语言理解团队的领导者。他的职业生涯始于东京工业大学的助理教授。1998 年和 2002 年分别加入了九州工业大学和奈良先端科技大学，担任副教授。2010 年，加入了日本东北大学担任教授。研究领域包括自然语言处理和人工智能。目前担任 EMNLP-IJCNLP 2019 的总主席，日本自然语言处理协会的理事，日本科学委员会的委员以及 NPO FactCheck Initiative Japan 的理事。

文献综述

张紫岩

2023.3.30

（苏州大学 计算机科学与技术学院）

1 引言

随着信息时代的到来，数据量正在以惊人的速度增长。人类对于自然语言处理的需求也越来越大，机器自动处理数据的方式逐渐取代了人工处理。作为自然语言处理的一项重要应用，文本纠错技术在日常生活中应用广泛，如文本校对工具、写作助手、协助人们进行第二语言的学习等。

本文面向科技文献写作领域，分析文本纠错任务的主流模型和不同数据增强技术的实现，并进行了比较，总结了各自的优点和缺点。最后，分析了文本纠错任务进一步的研究方向。

2 语法纠错模型

早期纠错系统主要采用基于规则的方法，针对特定的语法错误，制定对应的规则进行纠正。这种方法只能纠正特定的几种错误，且纠错效果完全依赖于制定的规则的好坏。为了解决基于规则的纠错方法的局限性，Brockett 等[1]首先将文本纠错任务看作翻译任务,并采用了基于统计机器翻译的方法构造纠错模型，通过字、词等相关信息的 n-gram 获取文本特征,对语言进行建模。但统计方法的泛化能力较差，目前也已经被淘汰。

近年来，基于神经网络的模型由于具有很强的自动特征学习能力，在自然语言处理领域中得到了广泛的应用。文本纠错在神经网络阶段主要采用的模型框架主要可以分为两类，序列到序列（Sequence-to-Sequence，Seq2Seq）模型以及序列到编辑（Sequence-to-Edit，Seq2Edit）模型。

2.1 Seq2Seq 模型

随着深度学习的出现，以及机器翻译和其他 Seq2Seq 任务中有前景的结果，神经机器翻译（Neural Machine Translation，NMT）自然被扩展到文本纠错任务。相对于传统的统计机器翻译（Statistical Machine Translation，SMT），NMT 使用单个大型神经网络来模拟整个纠错过程，消除了复杂的文本纠错特定特征工程的需求。尽管 NMT 很简单，但在各种纠错任务上已经实现了最先进的性能。Seq2Seq 模型包含两个基本的结构：编码器（Encoder）和解码器（Decoder）。通常而言，解码端是自回归进行输出的，即模型解码输出一个词时需要看到前面已生成的词。由于自回归解码性能较好，是目前 Seq2Seq 模型通用的解码方案。但是，由于自回归解码无法并行化，所以速度较慢。此外，传统的 Seq2Seq 模型编码器仅通过一个固定维度的向量来表示整个源端句子的语义信息，从而不可避免地导致了许多信息的丢失，随着输入句子长度的增加，此现象将会越来越严重，导致该结构的模型的性能显著下降。为了解决传统 Seq2Seq 模型存在的缺陷，谷歌的研究人员提出了 Transformer [2] 模型，通过多头自注意力机制较好地解决了上述问题。它不仅使编码器获得了并行编码的能力，加速了运算，还通过矩阵运算让句子中任意两个单词的距离变为 1，从而能够充分捕捉句子中的远距离依赖信息，完全摒弃了传统 Seq2Seq 模型的架构，获得了巨大的性能提升。

2.2 Seq2Edit 模型

虽然大多数文本纠错方法根据输入句子生成一个更正后的句子，但 Seq2Edit 模型生成一系列编辑，以应用于输入句子,如表 2-1 所示。由于纠错任务中从输入到输出的过程中大部分词元被直接复制，Stahlberg 和 Kumar[3]认为生成完整序列是浪费的。通过生成编辑操作而不是完整句子，Seq2Edit 模型通常具有更快的推理速度，比生成整个句子的纠错系统快 5 到 10 倍。然而这种方法的一个限制是编辑操作往往是基于词元的，因此有时无法捕获更复杂的多词元流畅性编辑。

因此，我们将 GEC 建模为一个 Seq2Seq 任务，并采用 Transformer 模型作为主模型。

3 数据增强技术

随着神经网络的不断发展，且模型的尺寸不断增加，人类对高质量数据的需求也在不断增长。为文本纠错模型收集大量训练数据一直是语法纠错领域面临的挑战。文本纠错最大的公开并行数据集 Lang-8 仅包含 200 万个句子对，其中中文语料更是稀缺。可用的数据量不足以支撑模型归纳出的各种语法错误。然而，获取人工标注数据既昂贵又困难，因此，从干净的单语语料库中生成合成平行语料库的技术得到了广泛的研究。通过在句子中加入杂音，并将其与原句进行配对，生成合成的平行语料库。合适的数据增强方法可以为中文语法纠错模型性能带来稳定的改善。目前，语法纠错数据增强的方法主要有直接噪声和反向翻译噪声。

3.1 直接噪声

人工合成语法错误的一种方法是干扰正确文本，使其在语法上不正确。这些干扰可以是基于规则的噪声操作或经常出现在纠错并行语料库中的错误模式。直接噪声所产生的结果在相同的数量级上更容易使用，因为它不需要训练语言模型，也不需要外部数据集。主要可以分为两种类型：基于规则的方法和基于规则的方法。（1）基于规则的方法：向一个干净的语料库添加噪声的最直观的方法是根据一些预定义的规则应用一系列干扰操作，将噪声直接加入到符合语法的句子中。Ehsan 和 Faili[4] 从预定义的错误模板中为每个句子添加一个错误，这些错误模板包括省略介词、单词重复等。Lichtarge 等[5]通过删除、插入、替换和交换字符位置，将拼写错误引入维基百科的编辑历史。（2）基于错误模式的方法：生成伪数据的另一种方法是注入文本纠错并行语料库中经常出现的错误。这种方法可以使得合成的错误就更符合自然界中真实错误的发生规律。Rozovskaya 和 Roth[6]提出了三种不同的文章错误注入方法，即分别根据纠正前文章的分布、纠正后的文章分布，文章被纠正部分的分布来添加错误。Felice 和 Yuan[7]后来对该方法进行了改进，在产生人工错误时考虑了文本的词法、POS 标签、语义概念和词义信息。Rei[8]进一步将其扩展到所有类型的误差。

3.2 反向翻译噪声

基于反向翻译的方法指的是将训练集中的数据翻转，训练一个从正确句子到错误句子的模型，然后用该模型对大规模单语正确语料加入噪音，再使用人造平行数

据对正向模型进行预训练，从而提升模型的鲁棒性。该技术最初被提出是为了在机器翻译中生成额外的数据，但它也可以直接应用于文本纠错。Rei 等[8]首次将反向翻译应用于语法错误检测，Yuan 等[9]在 Rei 等[8]的基础上增加了一种基于语言模型概率的质量控制，以确保生成的合成的伪数据比原始输入句子包含的语法错误更多。Kiyono 等[10]比较了反向翻译噪声和直接噪声，发现反向翻译噪声效果更好。目前，在机器翻译研究领域，由于反向翻译具有较强的性能，被认为是生成伪数据的标准方法。然而，对于文本纠错来说并非如此，单纯地应用反向翻译只会导致性能的微小改善，因为在原始的反向翻译中，模型通常过于保守，只能产生训练集中出现频率较高的错误，也就是说，反向模型同样会受到训练数据不足的影响。尽管引入预训练语言模型可以一定程度上缓解上述问题，但错误的多样性依旧不足。

3.3 领域词典约束的数据增强

综上所述，我们采用基于规则的数据增强方法。与常见的基于规则的方法不同，我们针对领域词典进行数据增强。具体来说，对于领域词典中的每个词元概率地选择替换、插入、交换、删除操作，而非随机选取句子中的任意词元进行上述操作。由于本语法纠错任务针对科技文献领域，对该领域中的高频关键词进行数据增强工作可以更有效地提升模型性能。

4 结语

中文语法纠错是如今中文自然语言处理领域的一个重要研究方向。比起英文语法纠错，中文的数据集更稀缺，如何获取高质量大规模数据成为了提升中文语法纠错模型性能的关键问题。另外，针对特定领域的语法纠错研究较少，且特定领域的真实语料规模更加有限，如何利用现有数据进行数据增强势必是一个重要的研究方向。因此，探讨领域词典约束下的语法纠错任务可以有效填补上述研究方向的空白。

参考文献

[1] Brockett C, Dolan W B, Gamon M. Correcting ESL errors using phrasal SMT techniques[C]. In Proceedings of ACL, 2006: 249-256.

[2] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[3] Stahlberg, Felix and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations[C]. In Proceedings of EMNLP, pages 5147–5159.

[4] Ehsan, Nava and Heshaam Faili. 2013. Grammatical and context-sensitive error correction using a statistical machine translation framework[J]. Software: Practice and Experience, 43(2): 187–206.

[5] Lichtarge, Jared, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction[C]. In Proceedings of the 2019 Conference of the North American Chapter of ACL, pages 3291–3301.

[6] Rozovskaya, Alla and Dan Roth. 2010b. Training paradigms for correcting errors in grammar and usage[C]. In Proceedings of NAACL, pages 154–162.

[7] Felice, Mariano and Zheng Yuan. 2014a. Generating artificial errors for grammatical error correction[C]. In Proceedings of ACL, pages 116–126.

[8] Rei, Marek, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns[C]. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 287–292.

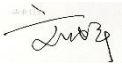
[9] Yuan, Zheng, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. Neural and FST-based approaches to grammatical error correction[C]. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 228–239.

[10] Kiyono, Shun, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction[C]. In

Proceedings of EMNLP-IJCNLP, pages 1236–1242.

苏州大学本科生毕业设计（论文）中期进展情况检查表

学院（部）：计算机科学与技术学院

<table>
<tr><td colspan="2">学生姓名</td><td>张紫岩</td><td>年级</td><td>2019</td><td>专业</td><td>计算机科学与技术（人工智能）</td><td>填表日期</td><td>2023/4/22</td></tr>
<tr><td colspan="2">设计（论文）选题</td><td colspan="7">领域词典约束的文本纠错技术研究与实现</td></tr>
<tr><td rowspan="2">已完成的任务</td><td colspan="8">得到领域词典和科技文献写作领域的数据，对数据进行清洗和加噪，完成基于 Seq2Seq 的文本纠错基线模型和基于领域词典的数据增强工作。对数据增强模型的实验结果进行分析总结，目前开始论文写作。</td></tr>
<tr><td colspan="2">是否符合任务书要求进度</td><td colspan="6">是</td></tr>
<tr><td rowspan="2">尚须完成的任务</td><td colspan="8">完成毕业论文写作</td></tr>
<tr><td colspan="2">能否按期完成任务</td><td colspan="6">是</td></tr>
<tr><td rowspan="2">存在的问题和解决办法</td><td>存在的问题</td><td colspan="7">1. 评价模型性能：标注的真实数据较少，其中包含的错误与数据增强得到的错误不符。</td></tr>
<tr><td>拟采取的办法</td><td colspan="7">1. 自动评价：测试集采用与训练集类似的数据增强方法。
2. 人工评价：随机采样一些数据进行人工评价。</td></tr>
<tr><td colspan="2">指导教师意见</td><td colspan="7">通过
签名：</td></tr>
<tr><td colspan="2">中期检查专家组意见</td><td colspan="7">同意
组长签名：</td></tr>
<tr><td colspan="2">学院（部）意见</td><td colspan="7">同意
教学院长签名：</td></tr>
</table>

检查日期：　　年　月　日

苏州大学本科生毕业设计（论文）答辩记录表

学院（部）：计算机科学与技术学院

学生姓名	张紫岩	年 级	2019	专 业	计算机科学与技术（人工智能）
设计（论文）题目	领域词典约束的文本纠错技术研究与实现				
答辩时间	2023 年 5 月 17 日 13：30 至 17：00		答辩地点	理工楼 243	
答辩小组成员： 王红玲，洪宇，陈伟，张正超					

答辩中提出的主要问题及学生回答问题的简要情况：

问：你用了两个不同的测试集吗？

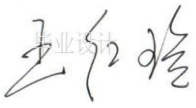
答：是的，因为本次数据增强实验采用了两种评价方式，自动评价和人工评价。自动评价的测试集采用 500 句人工标注的平行句子对和 500 句数据增强得到的句子对；人工评价从 CCF 文献数据集中采样 2 万句进行加噪，形成了一个较大规模测试集。

问：你是怎么实现数据增强的？

回答：本次毕业设计主要是提出了一种基于规则的数据增强方法，使用这种增强方法对单语语料库进行加噪，生成一批高质量纠错句子对，供模型训练。

问：你这个特定领域的文本纠错和普通领域的有什么区别？

答：不同于通用领域，特定领域的文本中包含领域内的知识与信息，如一些领域内的专业词汇，如果不将这些领域内的知识引入纠错模型，那么会影响模型判别错误以及纠正。本次毕业设计主要是利用领域词典将领域知识引入纠错模型。

答辩小组组长签名：

答辩小组成员签名：

记录人签名：

2023 年 5 月 17 日

苏州大学本科生毕业设计（论文）成绩评定表

学院（部）：计算机科学与技术学院

<table>
<tr><td colspan="4">设计（论文）题目：领域词典约束的文本纠错技术研究与实现</td></tr>
<tr><td>姓 名</td><td>张紫岩</td><td>学 号</td><td>1928401062</td></tr>
<tr><td>年 级</td><td>2019</td><td>专 业</td><td>计算机科学与技术（人工智能）</td></tr>
<tr><td>指导教师评语</td><td colspan="3">张紫岩同学的毕业内容是针对科技文献领域文本纠错问题，实现文本纠错系统，进而设计基于词典的数据增强方法、自动构造面向科技文献领域文本的训练数据集，最后进行实验对比。张紫岩同学学习态度很好、做事情有责任心，努力完成每一项任务，比较完满地完成了预定目标。论文书写规范、认真。</td></tr>
<tr><td></td><td colspan="3">
<table>
<tr><td>评 价 内 容</td><td>得 分</td></tr>
<tr><td>设计（论文）方案设计、文献检索、阅读及综述能力、进度等情况评价分（计 25 分）</td><td>23</td></tr>
<tr><td>毕业设计（论文）质量和工作量评价分（计 50 分）</td><td>46</td></tr>
<tr><td>科学素养、学习态度、纪律表现等情况评价分（计 25 分）</td><td>23</td></tr>
<tr><td colspan="2">成绩（满分 100 分）：92 签名： 2023 年 5 月 16 日</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>评阅教师评语</td><td colspan="2">张紫岩同学的论文描述了一个面向科技文献领域的领域词典约束的文本纠错技术的研究与实现，选题具有一定的理论研究价值，不仅使用了常用的模型来训练纠错模型，还使用了数据增强技术，具有一定的工作量。说明该生已具备了一定的分析问题及解决问题的能力。该生所写的毕业论文，结构合理，书写流畅，基本概念表述清楚，排版较规范。总体上，符合本科生毕业论文要求。
建议：表的标题应放在表格正上方，适当增加中文参考文献。</td></tr>
<tr><td rowspan="6">答辩小组评语</td><td>评 价 内 容</td><td>得 分</td></tr>
<tr><td>毕业设计（论文）文字书写评价分（计 20 分）</td><td>19</td></tr>
<tr><td>毕业设计（论文）质量评价分（计 40 分）</td><td>37</td></tr>
<tr><td>工作量情况评价分（计 20 分）</td><td>18</td></tr>
<tr><td>毕业设计（论文）创新及分析问题、解决问题能力评价分（计 20 分）</td><td>17</td></tr>
<tr><td colspan="2">成绩（满分 100 分）：91 签名：王红梅 2023 年 5 月 17 日</td></tr>
<tr><td rowspan="6">答辩小组评语</td><td colspan="2">张紫岩同学在答辩时能运用所学理论和专业知识，较全面地阐述了论文的主要内容，论点基本正确，语言表达较准确，表现出对所从事的研究内容掌握较好。回答问题时，有理论根据，思路较清晰，基本概念清楚，能较准确地回答提出的问题，表现出理论知识掌握较好。从论文内容看，较好地完成了任务书所规定的各项要求，论文书写符合规范。</td></tr>
<tr><td>评 价 内 容</td><td>得 分</td></tr>
<tr><td>毕业设计（论文）介绍表达情况评价分（计 20 分）</td><td>15</td></tr>
<tr><td>回答问题表现评价分（计 40 分）</td><td>35</td></tr>
<tr><td>毕业设计（论文）水平和工作量评价分（计 40 分）</td><td>35</td></tr>
<tr><td colspan="2">成绩（总分 100 分）：85 组长签名：王红梅 2023 年 5 月 17 日</td></tr>
<tr><td colspan="2">按权重折算总成绩</td><td>89</td></tr>
<tr><td colspan="3">审定成绩：89 答辩委员会主任签名：
日 期：</td></tr>
</table>

文本复制检测报告单

№:ADBD2023R_2018112714114520230530114445814401542299

检测文献： 领域词典约束的文本纠错技术研究与实现

作者： 张紫岩

检测范围：
中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
CNKI大成编客-原创作品库

时间范围： 1900-01-01至2023-05-30

检测时间： 2023-05-30 11:44:45

总文字复制比：4.6%

去除引用：4.6% **去除本人：4.6%** **重合字数：1008** **文献总字数：22129**

单篇最大文字复制比：1.4% **单篇最大重复字数：306**

总段落数： [3] 疑似段落数： [3] 疑似段落最大重合字数： [851]

前部重合字数： [474] 后部重合字数： [534] 疑似段落最小重合字数： [31]

8.4%	领域词典约束的文本纠错技术研究与实现_第1部分(总10148字)
1.3%	领域词典约束的文本纠错技术研究与实现_第2部分(总9822字)
1.4%	领域词典约束的文本纠错技术研究与实现_第3部分(总2159字)

中国知网文献检测报告单（机构版）