

SynGEC:语法增强的语法错误纠正 使用定制的面向 GEC 的解析器

匿名 ACL 提交

抽象的

尽管他们取得了巨大的成功,但目前
尖端的语法错误纠正
(GEC) 模型很少使用句法
知识,起着重要的作用
当人类试图理解和修正不符合语法的句子时。这项工作
提出
语法增强的 GEC 方法 (SynGEC),
将句法信息纳入
GEC模型的编码器部分。这个想法的关键挑战是

现成的解析器急剧下降,因为
他们通常接受干净的语法训练
句子。为了应对这一挑战,我们建议构建一个定制的
面向 GEC 的解析器
(GOPar) 使用并行 GEC 训练数据作为
一个支点。首先,我们提出了一个扩展的注释方
案,它允许我们同时表示
统一树形结构下的语法错误和句法结构。然后,我们得
到
解析源错误句子的树
通过投影目标正确句子的树并将它们用于训练 GOPar。
我们
使用图卷积网络进行编码
GOPar 生成的树结构。三个英文/中文基准的实验

数据集表明,我们提出的 SynGEC 方法始终如一且大
大优于强大的基线并实现了新的

单模型最先进的性能
所有数据集。

1 简介

给定一个可能不合语法的句子,
语法纠错 (GEC)任务旨在
生成语法目标句 (Grund kiewicz 等人, 2020; Wang 等
人, 2021)。最近的
主流方法将 GEC 视为单语言机器翻译 (MT) 任务 (Yuan 和
布里斯科, 2016; Junczys-Dowmunt 等人, 2018 年)。
基于标准编码器-解码器的 MT 模型,例如,
变压器 (Vaswani et al., 2017)已经出现
作为主导范式并在各种 GEC 数据集 (Rothe

等人, 2021;斯塔尔伯格和库马尔, 2021;孙等人,
2022;张等人, 2022)。尽管取得了令人印象深刻的成
就,但大多数作品将输入句子作为一系列标记,完全忽略
了内部句法或语义结构。

与 MT 相比,GEC 有两个特点直接推动了这项工作。首先,
GEC 模型的训练数据要少得多,并且可以通过结合语法等语
言结构知识来缓解数据稀疏问题。如表1和表4 所示,如果不考
虑高噪声人群注释的 Lang8 数据,英文和中文 GEC 任务只有
大约126K 和 157K 高质量标记的源/目标句对进行训练 (Mita
et al., 2020)。其次,不合语法的句子中的许多错误本质上与
句法解析相关。例如,一些错误,例如时态或单复数形式的不一
致,可以在远程句法依赖的帮助下更好地检测和纠正。

在本文中,我们提出了 SynGEC,一种将输入句子的句法
结构注入 GEC 模型的编码器部分的方法。这个想法的关键挑
战是对不合语法句子的解析性能不令人满意。通常,现成的句
法解析器是在干净的树库上训练的,而 GEC 的输入句子通常
包含各种语法错误。如此大的差距不可避免地会导致许多格
式错误的解析树 (Hashemi and Hwa, 2016)。据我们所
知,最近的两项工作试图将句法知识纳入 GEC,但都未能直接
应对这一挑战 (Wan and Wan, 2021; Li et al., 2022)
(有关更详细的讨论,请参见第6节)。

同时,有一系列工作试图通过扩展注释指南和手动注释数
据来适应不合语法文本的句法解析 (Dickinson 和
Ragheb, 2009; Berzak 等, 2016; Nagata 和
Sakaguchi,

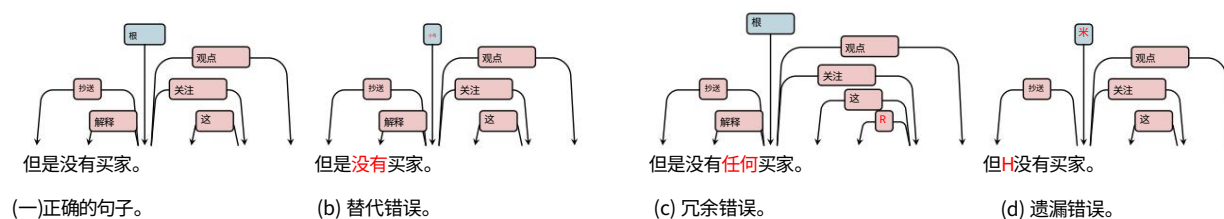


图 1:我们针对所有三种语法错误的扩展注释方案的图示。

2016)。但是,手动注释非常

昂贵,因此现有的注释数据集是

规模很小。

面对现成的解析器在不合语法上性能不可靠的挑战

句子,我们建议训练一个定制的面向 GEC 的解析器

(GPar)。基本思想是在

GEC 训练数据。首先,我们使用普通的现成解析器解析目标正确的句子。

然后,我们通过树投影为不正确的句子构建树。为了适应语法错误,我们提出了一个基于几个简单规则的扩展注释方案,

这允许我们在一个统一的树中表示语法错误和句法结构。最后,

我们直接在GEC中自动构建的源语句树上训练 GPar

训练数据。

为了合并句法信息,我们级联了几个图卷积网络 (GCN)

编码器上方的层 (Kipf 和 Welling, 2017)

我们的基线基于 Transformer 的 GEC 模型。

我们在两个广泛使用的英语 GEC 评估数据集上进行实验,即 CoNLL-14 (Ng

等人, 2014 年)和 BEA-19 (布莱恩特等人, 2019 年),以及

最近提出的中国 MuCGEC 数据集

(张等人, 2022)。结果表明,我们的 Syn GEC 方法实现了一致和实质性

改进所有数据集,即使使用 BART 增强了基线模型 (Lewis 等人,

2020)。SynGEC 实现新的单模型 SOTA

结果:在 CoNLL-14 上的F0.5得分为 69.6, F0.5为 72.9

BEA-19得分,MuCGEC 得分46.8 F0.5,

在可比条件下优于之前的 SOTA

大幅设置 (+3.5/+0.5/+7.3)。我们将

在<http://github.com>上发布我们的代码。

2 我们的面向 GEC 的解析器

本节介绍了我们量身定制的 GPar,它

在解析不合语法的句子方面比现成的解析器更有能力。

2.1 扩展注释方案

表示句法的标准注释方案是为语法句子设计的,因此不涵盖语法错误句子中的许多结构。因此,为了获得一个定制的解析器,我们的首要任务是扩展注释方案,更具体地说,设计一套补充的注释指南来处理不同的语法错误。通过这种方案,我们可以使用统一的树结构来表示语法错误和句法信息。

如图1所示,我们提出了一个基于几个直接规则的简单扩展注释方案,对应于三种类型的语法错误,即替代、冗余和缺失 (Bryant 等人, 2017)。相应地,我们在原始句法标签集中添加三个标签,即“S”、“R”和“M”,以捕获三种错误。由于语法错误检测 (GED)任务 (Yuan et al., 2021)也采用了这种分类,我们将它们称为 GED 标签。

替代错误 (S)包括拼写错误、时态错误、单数/复数不一致错误等。为简单起见,我们不考虑这种细粒度的类别,并使用单个“S”标签表示该单词应替换为另一个,如图1(b)所示。

冗余错误 (R)意味着应该删除一些单词。对于每个冗余词,我们让它依赖于它的右侧相邻词,并带有标签“R”,即使右侧相邻词也是冗余的,如图1(c)所示。

缺失错误 (M)意味着应该插入一些单词。对于每个缺失的单词,我们为其右侧相邻单词的传入弧分配一个标签“M”,除非缺失的单词位于句子的末尾,如图1(d)所示。

如果缺少几个连续的单词,则注释与缺少单个单词时保持相同。此外,由于一个缺失的单词可能在正确句子的树中有孩子,我们让它们依赖于缺失的头词

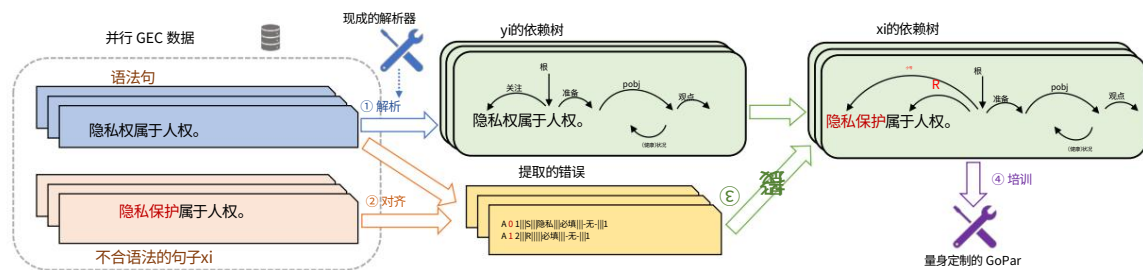


图 2:获取我们定制的 GOPar 的工作流程。

词,而不改变它们的句法标签。

基于上述简单的规则,我们的目标是进行最小的调整并做出结

果

树尽可能与句法树一致

的正确句子。同时,请

注意我们的注解方案可能会遇到

不同类型的错误发生时的问题

句子的连续位置。采取“但是

没有买家”为例,我们需要更换

“was”和“were”,然后在前面插入“there”

同时“是”。因此,根据

我们的规则,传入弧的标签“是”

可以是“S”或“M”,导致标签冲突。要确定一个唯

一的标签,我们只需定义一个

优先顺序:“S”±“R”±“M”。

2.2 构建 GOPar

基于上述标注方案,我们提出

使用并行 GEC 训练我们定制的 GOPar

训练数据D” tpxi, yiqu作为支点。主要目标是自动生成高质量的

用真实的大型句子解析树

语法错误,并使用它们来训练解析器

适合解析不合语法的句子。图2说明了工作流程,包括

以下四个步骤。

首先,我们使用现成的解析器来解析

瞄准 GEC 训练数据的正确句子(例如, yi)。由于这样的

句子没有语法错误,解析器更有可能产生

可靠的解析树。

其次,我们使用 ERRANT (Bryant et al., 2017)¹

根据对齐提取源错误句子(例如, xi)中的所有语法

错误

在xi和yi之间第三,我们通

过将yi的树投影到任何错误、依赖关系(弧)和标签来构造xi的解析

树。对于不相关的词

直接复制;对于那些与错误有关的,取决于

¹<https://github.com/chrisjbryant/>

错误的

密度和标签是根据第2.1节介绍的规则分配的。

第四,为 D 中的所有源端句子构建解析树,然后我们将它们用作树库来训练我们定制的 GOPar。

构建定制 GOPar 的另一种方法是直接使用手动标记的树库。由于现有的树库只包含语法句子,我们可以根据规则或反向翻译模型注入合成错误 (Foster et al., 2008; Cahill, 2015)。然后,我们可以通过上述第二步和第三步类似地为不合语法的句子生成解析树。然而,我们的初步实验表明,以这种方式构建的 GOPar 要差得多,只能稍微改进我们的基线 GEC 模型。我们怀疑原因有两个。一方面,合成语法错误与真实语法错误之间存在相当大的差距;另一方面,由于现有树库的规模有限,生成的数据不足以充分训练 GOPar。

3 基于 DepGCN 的 GEC 模型

在本节中,我们将描述我们基于 DepGCN 的 GEC 模型。受 Wan and Wan (2021) 的启发,我们首先采用 GCN (Kipf and Welling, 2017)对源句的依存句法树进行编码。

然后,我们将编码的句法信息输入到基于 Transformer 的 GEC 模型中。

3.1 变压器骨干

我们将 GEC 建模为序列到序列的任务,并采用常用的 Transformer 模型(Vaswani et al., 2017)作为主干。变压器由编码器和解码器组成。

编码器利用多头自注意力机制来获取源句中每个标记的上文表示。这

解码器具有类似的架构,同时还包含一个带掩码的多头自注意力模块来对生成的令牌信息进行建模。

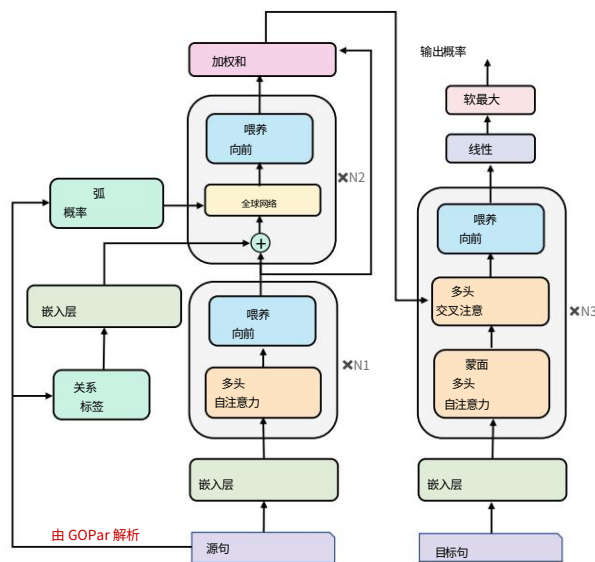


图 3: 我们基于 DepGCN 的 GEC 模型概述。
操作 表示向量连接。N1, N2
和 N3 表示相同的 Transformer 编码器块、DepGCN 块和
Transformer 解码器的数量
块, 分别。我们还采用了残差连接(He et al., 2016), 然后是层
归一化
(Ba et al., 2016) 围绕每个子层。

在训练期间, 目标函数是最小化教师强制负对数似然

损失 (Williams 和 Zipser, 1989), 正式地:

$$L_{\theta} = -\sum_{i=1}^n \log P(y_i | x; \theta) \quad (1)$$

“我们的兴趣”
我 “1”

其中 θ 是可训练的模型参数, x 是
源句, $y = (y_1, y_2, \dots, y_n)$ 是具有 n 个标记的 ground
truth 目标句, 并且 y_i 是

y_1, y_2, \dots, y_i 是第 i 个训练时间步的可见标记。

在推理过程中, 我们利用波束搜索
解码 (怀斯曼和拉什, 2016) 找到一个
最优序列 \hat{y} 通过最大化条件
概率 $P(\hat{y} | x; \theta)$ 。

以前的工作表明 PLM, 例如 BART
(Lewis 等人, 2020) 和 T5 (Raffel 等人, 2020),
可以大幅度提高 GEC 性能
(Rothe 等人, 2021 年; Sun 等人, 2022 年)。我们进一步
使用 BART 建立更强大的基线, 因为它共享
与我们的 Transformer 相同的模型架构
骨干。具体来说, 我们使用 BART 参数来初始化我们的
Transformer 主干网络和

然后继续对 GEC 训练数据进行训练。更多细节在第 4.1 节和
第 5 节中讨论。

3.2 依赖 GCN (DepGCN)

我们使用 GCN 对依赖语法信息进行编码, 命名为 DepGCN。
DepGCN 模块堆叠了几个相同的块, 每个块都有一个 GCN 子
层和一个前馈子层。

对于 GCN 子层, 我们引入了依赖弧和依赖 plq 的信息

同时贴标签。我们计算输出 h
在第 i 个令牌处的第 l 个 GCN 为:

$$H_{plq}^i = \sum_{j=1}^n A_{ij} W_{plq} h_j^{l-1} \quad (2)$$

其中 $A \in \mathbb{R}^{n \times n}$ 表示邻接矩阵, 是依赖标签的嵌入, 并
且
词 w_i 和词 w_j 之间, $W_{plq} \in \mathbb{R}^{d \times d}$
是模型参数。ReLU (Nair and Hinton, 2010)
是激活函数, 指的是向量连接。

为了减少 GOPar 的错误传播, 请遵循 Zhang 等人。
(2020a), 我们使用从 GOPar 导出的弧概率矩阵作为邻接矩
阵 A , 它可以提供所有潜在的句法结构。

然后, 我们将 GCN 子层的输出馈送到前馈 (FF) 层,
该层包含两个线性变换和 ReLU 激活函数

之间, 如下图:

$$FF(h) = \text{ReLU}(W_1 h + b_1) W_2 + b_2 \quad (3)$$

3.3 整合句法信息

我们将 DepGCN 放在 Transformer 编码器和解码器之间的
连接处, 并构建了一个基于 DepGCN 的 GEC 模型。如图 3 所
示, 给定一个输入句子 x 和相应的句法信息 x

syn (包括依赖弧和标签), 我们首先从 Transformer 编码
器中获得 x 的上下文表示 H_c 。然后, 我们输入 H_c 和 x

syn 进入 DepGCN 并获得语法
增强的表示 H_{syn} 。
最后, 我们使用加权求和运算推导出融合表示 H_f 作为

变压器解码器:

$$H_f = \beta H_c + (1 - \beta) H_{syn} \quad (4)$$

其中 $\beta \in [0, 1]$ 是一个超参数。训练和推理过程与
Transformer 基线一致。

数据集	#Sentences	%错误	用法
CLang8 2,372,119		57.8	预训练
FCE 34,490		62.6	微调我
核57,151		38.2	微调我
W&I+LOCNESS 34,308		66.3	微调 I&I
BEA-19-Dev 4,384		65.2	验证
CoNLL-14-测试1,312		72.3	测试
BEA-19-测试4,477		-	测试

表 1:英语 GEC 数据集的统计数据。

4 英语GEC实验

4.1 设置

数据集和评估。我们首先预训练我们的在 Lang8 数据集的清理版本上建立模型 (CLang8) 由 Rothe 等人发布。(2021 年)。然后, 我们使用 FCE 数据集 (Yannakoudakis 等人, 2011), NUCLE 数据集 (Dahlmeier 等人, 2013) 和 W&I+LOCNESS 训练组 (Bryant 等人, 2019)用于模型微调学习。像 Omelianchuk 等人一样。(2020),我们将微调过程分解为两个阶段:

- 1)对FCE+NUCLE+W&I+LOCNESS进行微调;
- 2)仅在小范围内进一步微调,但高品质的 W&I+LOCNESS。为了评估,我们报告三个不同运行的平均结果

CoNLL-14 测试集(Ng³ et al., 2014)由 M2Scorer (Dahlmeier 和 Ng, 2012 年)和 BEA-19 由 ERRANT 评估的测试集(Bryant et al., 2019) (布莱恩特等人, 2017)。BEA-19 开发集服务于作为整个训练期间的验证数据。这上述数据集的统计数据如表1所示。

GEC 模型详细信息。我们采用 Fairseq4 (那里 et al., 2019)来构建我们的 Transformer 基线和基于 DepGCN 的模型。对于基于 DepGCN 在模型中,我们堆叠 3 个 DepGCN 块,并将公式4中的聚合因子 β 设置为 0.5。我们应用 BPE (Sennrich et al., 2016)生成 32K 共享子词词汇表。我们重新实现了 Junczys-Dowmunt 提出的 Dropout Src 机制

等。(2018)并在源词上执行嵌入矩阵以减轻过拟合。更多的模型细节在附录A中讨论。

GOPar 详细信息。我们使用双仿射解析器 (Dozat and Manning, 2017)作为 GOPar 的模型结构。为了实施,我们采用解析工具包 Supar5 (Zhang 等人, 2020b)和

2CLang8 可以从<https://github.com/google-research-datasets/clang8> 下载。

3我们使用official-2014.combined-withalt.m2版本的 CoNLL-14 遵循之前的 SOTA (Rothe 等人, 2021)。

4<https://github.com/pytorch/fairseq>
5<https://github.com/yzhangcs/parser>

在他们的论文之后设置超参数。在比较了几个流行的 PLM 之后,我们使用 ELECTRA (Clark et al., 2020)来提供上下文令牌表示。

GOPar 训练数据是通过第 2 节中详细描述的数据构造方法从 CLang8 GEC 数据集 (Rothe 等人, 2021)生成的。对于在 GEC 中使用 GOPar,由于 GEC 模型的输入是子词序列,而基本句法树中的标记单元是单词,如果在将该树输入 GEC 模型之前,在单词级树中存在从单词 w 到单词 v 的弧,我们将 w 的所有子单词的弧添加到 v 的所有子单词。

纳入 BART。为了验证在结合强大的 PLM 后句法知识仍然有用,我们使用 BART (Lewis 等人, 2020)来初始化我们 mod- 的 Transformer 主干。

其他。值得注意的是,在将BART 结合到我们的基于 DepGCN 的模型中时,我们采用了两阶段的训练程序来保持训练的稳定。首先,我们微调 BART 初始化的 Transformer 主干,直到它收敛。其次,我们在融合的 Transformer 主干中添加了一个辅助 DepGCN 模块,并且只在相同的训练数据上调整 DepGCN 参数。

4.2 主要结果

主要结果列于表2。在没有 PLM 的第一组结果中,我们的 SynGEC方法在 CoNLL -14 和 BEA-19 测试集上分别获得 65.2/68.4 F0.5分数,优于所有其他使用语法的系统。SynGEC在两个测试集上的性能仅低于Stahlberg 和 Kumar (2021),因为他们使用了 5.4 亿个句子的超大型合成语料库,而我们只使用了 240 万个句子。GOPar 提供的句法信息的结合导致比我们的基线提高3.5/4.2 F0.5,这表明来自GOPar 的句法知识对 GEC 非常有帮助,我们基于 DepGCN 的 GEC 模型可以有效地捕获它。

在底部组中,我们使用 BART 增强的 SynGEC 方法获得了 69.6/72.9 F0.5分数,在类似规模下持续优于其他 PLM 增强模型7

· 去除语法后,两个数据集的F0.5分数下降了 0.9,

6我们实验中使用的所有PLM的下载链接可以在附录B中找到。

7 Rothe 等人。(2021)还建立了一个更大的 GEC 模型在 T5-11B 上 (Raffel 等人, 2020)并达到 68.9/75.9 F0.5 分数。为了公平比较,我们没有在表2作为这个模型大约比我们的大 24 ^。

		额外的	变压器	CoNLL-14-测试数	BEA-19-测试	
系统无语		据大小层,隐藏,FFN PR F0.5 PR F0.5				
无 PLM	法					
	Kiyono 等人(2019)	70M	12+12,1024,4096	67.9 44.1 61.3 65.5 59.4 64.2		
	Lichtarge 等人。(2020) $\triangle \blacktriangle$	340M	12+12,1024,4096	69.4 43.9 62.1 67.6 62.5 66.5		
	斯塔尔伯格和库马尔(2021) $\triangle \blacktriangle \square$	540M	12+12,1024,4096	72.8 49.5 66.6 72.1 64.4 70.4		
	我们的基线 \heartsuit w/ 语	2.4M	6+6,512,2048	69.6 42.4 61.7 66.8 55.5 64.2		
	法					
	万和万(2021) \blacklozenge		6+6,512,2048	74.4 39.5 63.2 74.5 48.6 67.3		
	10MLI等人。(2022) \clubsuit 30M SynGEC \heartsuit		12+12,1024,4096	66.7 38.3 58.1 71.6 - - -		
		2.4M	6+6,512,2048	47.9 65.2 70.9 59.9 68.4		
	GOPar \tilde{N} 现成的解析器2.4M		6+6,512,2048	70.5 42.6 62.4 67.3 55.4 64.5		
带 PLM	无语法					
	Kaneko 等人(2020)	70M	12+12,1024,4096	69.2 45.6 62.6 67.1 60.1 65.6		
	Katsumata and Komachi (2020) \diamond	-	12+12,1024,4096	69.3 45.0 62.6 68.3 57.1 65.6		
	Omelianchuk et al. (2020)	9M	12+0,768,3072	77.5 40.1 65.3 79.2 53.9 72.4		
	罗特等人。(2021) \heartsuit	2.4M	12+12,1024,4096	- - 66.1 - - -		
	我们的基线 \heartsuit w/ 语	2.4M	12+12,1024,4096	75.6 50.4 68.7 74.0 64.9 72.0		
	法					
	李等人。(2022) \clubsuit	30M	12+12,1024,4096	68.1 44.1 61.4 - - -		
	SynGEC \heartsuit	2.4M	12+12,1024,4096	76.6 51.1 69.6 75.1 65.5 72.9		
	GOPar \tilde{N} 现成的解析器2.4M		12+12,1024,4096	76.1 50.0 68.9 74.6 64.1 72.3		

表 2:英语 GEC 测试集的单模型结果。我们的结果是在三个不同的运行中取平均值
随机种子。Layer、Hidden和FFN表示 Transform 的深度、隐藏大小和前馈网络大小。 “w/PLM”意味着使用预训练的语言模型。
“w/ syntax”意味着使用句法知识。
除了公开的人工注释训练数据外,当前的 GEC 系统还使用私有和/或人工
数据,包括: \heartsuit 人工 Gigaword (70M 句子), \triangle Wikipedia 修订历史 (170M), \blacktriangle artificial Wikipedia
(170M)、 \square 人工超大清洁爬虫语料 (200M)、 \heartsuit 非公开CLC (2M)、 \diamond 人工十亿字
(9M), \blacklozenge 人工十亿字 (10M), \clubsuit 人工十亿字 (30M), \heartsuit Lang8的清理版 (2.4M)。

这表明自适应的贡献
语法和 PLM 不完全重叠。

4.3 分析与讨论

GOPar 的有效性。在表2 中,我们提出
使用现成解析器的结果8
法知识 (GOPar \tilde{N} 现成的
解析器)。更改解析器后,影响
语法在所有设置下都变得边缘化。这个
观察表明,语法带来的性能提升很大程度上取决于
解析的质量。我们进一步研究解析
并发现 GOPar 在面对时更健壮
语法错误,并且可以进一步识别此类错误,而现成的
解析器易受攻击并且
倾向于提供不正确的解析。所以我们可以画
解析器的任务适应的结论是
将语法应用于 GEC 任务时必不可少。
句法信息的分解。至
获得更多关于形成自适应句法如何工作的见解,我们将其分
解为三个部分:
1)弧信息,这意味着只使用
语法树的拓扑结构; 2)
GED标签信息,指的是特殊的
标签 “S”、“R”和 “M”用于标记错误

8我们使用biaffine-dep-roberta-en Supar 提供的模型。

代币; 3) 句法标签信息,如不同句法关系的 “subj”和
“obj”。

我们进行消融研究以探索各种信息对 GEC9的影响
如表 3 所
示。首先,删除 GED 标签或语法标签会在类似程度上降低
SynGEC模型的性能,这表明它们对 GEC 同样重要。其次,当
我们只使用弧信息时,即去除所有标签时,与基线相比,召回率
急剧下降,而精度显着提高。我们推测弧信息的贡献主要是防
止 GEC 模型被不适当的上下文误导。第三,利用所有三个信息
的完整 SynGEC 模型实现了最佳性能,这意味着它们具有内
在的互补优势。

错误类型的表现。图4显示了BEA-19-dev 上不同
错误类型的更细粒度的评估结果。结果支持来自
GOPar 的句法信息对大多数错误类型都是有益的。
具体来说,句法知识

⁹对于 “w/o GED Labels”,我们强制解析器跳过 GED
标签并选择概率最高的语法标签
预测时。对于 “w/o Syntax Labels”,我们替换所有
结果中带有 “O”的语法标签。对于 “w/o All Labels”,我们
不要将标签嵌入输入到 DepGCN 模块中。

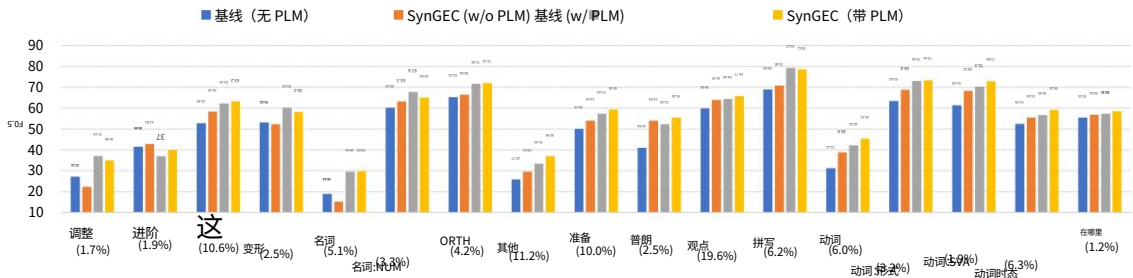


图 4:在 BEA-19-dev 上从 ERRANT (Bryant 等人, 2017 年)中选择的错误类型的性能。数字
括号中代表错误类型的百分比。我们排除了不到 1% 的错误类型。

	BEA-19-dev	CoNLL-14-测试
	P/R/F0.5 无	P/R/F0.5
PLM		
SynGEC	60.84/39.76/55.01	71.62/47.92/65.17
无GED 标签	59.47/38.03/53.44	72.28/45.43/64.64
无语法规则	59.31/39.02/53.72	71.61/46.65/64.69
不带所有标签	61.62/34.21/53.11	73.16/41.64/63.54
基线	57.99/35.77/51.58	69.60/42.37/61.67
带 PLM		
SynGEC	64.51/45.73/59.62	76.59/51.09/69.64
不带 GED 标签	63.59/45.23/58.82	76.79/50.45/69.53
无语法规则	64.20/45.51/59.33	76.32/50.91/69.39
不含所有标签	64.90/42.57/58.74	77.01/48.82/69.04
基线	63.09/44.80/58.32	75.61/50.42/68.74

表 3:我们方案中不同标签的效果。

数据集	#Sentences %错误用法
朗8	1,220,906 89.5 培训
汉语水平考试	15,6870 60.8 培训
MuCGEC-Dev 1,125 验证	95.1
MuCGEC-Test 5,938 测试	92.2

表 4:中国 GEC 数据集的统计数据。

PLM 语法 PR F0.5		
张等人。(2022) ✓ 基线	43.81	28.56 39.58
SynGEC	43.79	25.93 38.49
	✓ 46.88	27.68 40.58
基线	✓	54.21 28.51 45.93
合成GEC	✓	✓ 55.59 28.72 46.83

表 5:MuCGEC 测试的单模型结果。

显著提高 GEC 模型的能力
纠正上下文感知错误,例如 DET、PREP、
PUCNT、动词 :SVA 和动词 :时态。纠正此类错误需要远程信息,
哪些语法可以有效地提供。语法
还有助于解决字序 (WO) 错误,这
需要结构信息来纠正。此外,
在 PRON、OTHER、VERB 上的表现也是
大大改善。同时,我们注意到一个
一小部分类型受到负面影响,例如
ADJ、MORPH、SPELL 和名词。经过更多
仔细观察,我们发现他们的修正
主要依赖于本地信息,其中句法知识可能无济于事。

5 中国 GEC 实验

数据集和评估。我们直接按照
张等人最近的工作。(2022),谁提出了一个
多参考多源中文 GEC 评价
名为 MuCGEC 的数据集。我们报告P/R/F0.5
使用基于官方字符的评估工具进行 MuCGEC 测试的值。
MuCGEC-dev 用于
超参数调优。对于训练数据,我们使用
中国 Lang8 数据集(Zhao et al., 2018)和
HSK 数据集(Zhang, 2009)。以上统计

提到的数据集如表4 所示。
基于字符的 GOPar。当前的中文 GEC 模型通常将输入句子视
为字符序列,不进行分词。相反,句法解析器通常将输入句子视为单
词序列。为了处理这种不匹配,我们遵循Yan 等人。(2020)并构建
一个基于字符的 GOPar。基本思想是

将基于单词的树转换为基于字符的树
通过让每个 char 依赖于它在 multi-char 单词中的右边边。

使用 BART。我们采用了最近提出的中国 BART (Shao et al.,
2021),它最初是使用 HuggingFace Transformers 工具包 10
(Wolf et al., 2020) 实现的。我们设法包装他们的代码并将其用于
我们的 Fairseq实现,这比原始实现更快。更重要的是,我们发现它
的词汇表中缺少许多常用字符,因此根据我们的初步实验,添加了
3,866 个汉字和来自中文 Gigaword 和 Wikipedia语料库的标点
符号,从而显著提升了性能。

结果如表5 所示。

10<https://huggingface.co/>

使用 BART,我们的 SynGEC在 MuCGEC测试中以 2.09 F0.5的分数优于 Trans 前基线。使用 BART 时,我们的基线已经超越了之前的 SOTA 系统 (Zhang 等人, 2022), SynGEC 进一步提高F0.5分数 0.90。这些结果表明我们提出的 SynGEC 方法对于不同的语言可以稳定有效。

6 相关作品

语法错误更正。近期作品

主要将 GEC 制定为单语翻译任务,并以基于编码器-解码器的方式处理

MT 模型 (Yuan 和 Briscoe, 2016; Junczys Dowmunt 等, 2018),其中 Transformer (Vaswani et al., 2017)已成为主导范例。在合成训练数据的帮助下 (Lichtarge 等人, 2019 年; Yasunaga 等人, 2021 年)和大型 PLM (Kaneko 等人, 2020 年; Katsumata 和 Komachi, 2020),基于 Transformer 的 GEC 模型在各种

基准数据集 (Rothe 等人, 2021; Stahlberg 和库马尔, 2021)。

同时,序列到编辑 (Seq2Edit)方法作为一种竞争性替代方法出现,它

预测一系列编辑操作以实现

校正 (Gu et al., 2019; Awasthi et al., 2019; Omelianchuk 等人, 2020)。虽然这部作品采用基于 Transformer 的 GEC 模型作为基线,我们的 SynGEC 方法也可以应用

直接到 Seq2Edit 模型,我们留给以后的工作。

解析不合语法的句子。尽管

句法分析在干净句子上的成功

(Dozat 和 Manning, 2017; Zhang 等人, 2020b),当面对不合语法时,现成的解析器的性能会急剧下降

句子 (Hashemi 和 Hwa, 2016 年)。以前的研究主要通过注释小规模不合语法的句子并微调现成的解析器来解决这个问题 (Dickinson

和拉赫布, 2009;彼得罗夫和麦克唐纳, 2012;卡希尔, 2015; Berzak 等人, 2016)。相比之下,我们建议自动训练定制的解析器

从并行 GEC 数据生成语法树,这避免了费力的手动注释。

语法增强的 GEC。很多以前的作品

已经证明了利用的有效性

各种 NLP 任务的句法信息,例如

作为机器翻译 (Bastings et al., 2017; Zhang et al., 2019),意见角色标签(Zhang et al.,

2020a)和语义角色标签(Xia et al., 2019; Sachan et al., 2021)。同时,我们发现了两篇关于语法增强 GEC 的最新作品 (Wan and Wan, 2021; Li et al., 2022)。这两项工作都使用现成的解析器直接生成输入句子的依赖树,而无需为不合语法的句子定制解析器。他们都使用图注意力网络 (GAT) 进行树编码(Velickovic et al., 2018)。除了依赖树, Li 等人。 (2022)也利用了输入句子的构成树。

与上述两项工作相比,我们工作的主要贡献是通过我们定制的 GOPar直接处理严重的性能下降问题。我们采用 GCN 进行树编码是因为我们的初步实验表明它实现了相似的性能但速度更快。此外,我们的基线 GEC 模型比他们的模型实现了更高的性能,如表2 所示。

7 结论

本文提出了一种 SynGEC 方法,该方法有效地将句法知识整合到 GEC 模型中。关键思想是调整 vanilla解析器以适应不合语法的句子。

为了实现这一目标,我们首先扩展了标准句法注释方案,以使用统一的树结构来编码语法错误和句法结构。然后我们通过将目标句子树投影到 GEC 训练数据的源端来获得高质量的不合语法句子解析树,用于训练我们定制的 GOPar。为了结合GOPar 生成的解析树,我们提出了一个基于 Transformer 的基于 DepGCN 的 GEC 模型。在三个英文/中文数据集上的实验表明,我们提出的 SynGEC 方法是有效的,并且在可比较的设置下取得了新的 SOTA 结果。

限制和未来的工作。我们观察两个

我们目前工作的可能限制。首先,现成的解析器可能仍然会为 GEC 训练数据中的目标正确句子生成嘈杂的解析树,这进一步导致我们的源句子投影树中的噪声。其次,到目前为止,我们只使用三个粗粒度标签来区分注释方案中的语法错误,而细粒度类别可能会进一步使 GEC 受益。正如上面所讨论的,这两个限制都可以通过整合以前关于为不合语法的句子手动注释句法树的工作所获得的想法和资源来加强。

584	参考	蒂莫西·多扎特和克里斯托弗·D·曼宁。2017年。	637
585	Abhijeet Awasthi,Sunita Sarawagi,Rasna Goyal、	用于神经依赖解析的深度双仿射注意力。在 ICLR 会议记录中。	638
586	Sabyasachi Ghosh 和 Vihari Piratla。2019. 用于局部序列转导		639
587	的并行迭代编辑模型。在 EMNLP-IJCNLP 会议记录中,页	Jennifer Foster,Joachim Wagner 和 Josef Van Genabith。2008.	640
588		使 wsj 训练的解析器适应语法嘈杂的文本。在 ACL 会议记录 (短)中,	641
589	4260–4270。	第 221-224 页。	642
590	Lei Jimmy Ba,Jamie Ryan Kiros 和 Geoffrey E. Hin ton。2016.		643
591	层规范化。arXiv 预印本	Roman Grundkiewicz,Christopher Bryant 和 Mariano Felice。	644
592	arXiv:1607.06450。	2020. 自动语法纠错速成课程。在 COLING 会议录中:教程摘要,第	645
593	Jasmine Bastings,伊万·蒂托夫,威尔克·阿齐兹,迭戈	33-38 页。	646
594	马尔凯贾尼和哈利勒·西马安。2017. 图表		647
595	用于语法感知神经机器翻译的卷积编码器。在 EMNLP 会议记录中,	Jiatao Gu, Changhan Wang, and Jake Zhao Junbo。2019.	648
596	页面	莱文斯坦变压器。在 NIPS 会议记录中,第 11181-11191 页。	649
597	1957 年至 1967 年。		650
598	塞缪尔·贝尔、海伦·雅纳库达基斯和马克·雷克。	Homa B Hashemi 和丽贝卡·华。2016. 对不合语法句子的解析器鲁棒	651
599	2019. 上下文是关键:语法错误检测	性的评估。在 EMNLP 会议记录中,第 1765-1774 页。	652
600	与上下文词表示。在诉讼中		653
601	BEA@ACL,第 103-115 页。	Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian	654
602	叶夫根尼·贝尔扎克、杰西卡·肯尼、卡罗琳·斯帕丁、	Sun。2016. Identity mappings in deep residual net-	655
603	Jing Xian Wang, Lucia Lam, Keiko Sophie Mori,	works. In Proceedings of ECCV, pages 630–645.	656
604	塞巴斯蒂安加尔萨和鲍里斯卡茨。2016. 通用	Marcin Junczys-Dowmunt,Roman Grundkiewicz,Shubha Guha	657
605	学习者英语的依赖。在诉讼中	和 Kenneth Heafield。2018. 将神经语法错误纠正作为一项低资	658
606	ACL,第 737-746 页。	源机器翻译任务。在 NAACL -HLT 会议记录中,第 595-606 页。	659
607	Christopher Bryant,Mariano Felice,Øistein E Ander sen 和 Ted		660
608	Briscoe。2019. bea-2019分享	Masahiro Kaneko 和 Mamoru Komachi。2019. Multi -head	661
609	语法纠错任务。在诉讼中	multi-layer attention to deep language representations	662
610	BEA@ACL,第 52-75 页。	for grammatical error detection。计算研究资料库,第	663
611	克里斯托弗·布莱恩特、马里亚诺·费利斯和泰德·布里斯科。	883-891 页。	664
612	2017. 错误的自动注释和评估		665
613	语法错误纠正的类型。在 ACL 会议记录中,第 793-805 页。	Masahiro Kaneko,Masato Mita,Shun Kiyono,Jun Suzuki 和	666
614		Kentaro Inui。2020. 编码器-解码器模型可以从语法纠错中	667
615	奥菲·卡希尔。2015. 解析学习者文本:to shoehorn	的预训练掩码语言模型中受益。在 ACL 会议记录中,第	668
616	或者不给鞋拔子。在语言学学报	4248-4254 页。	669
617	注释研讨会,第 144-147 页。		670
618	Kevin Clark,Minh-Thang Luong,Quoc V. Le 和	胜侯悟和小町守。2020。	671
619	克里斯托弗·D·曼宁。2020. ELECTRA:预训练文本编码器作为鉴	使用预训练的编码器-解码器模型进行语法错误校正的更强基线。在	672
620	别器而不是	AACL 会议记录中,第 827-832 页。	673
621	发电机。在 ICLR 会议记录中。		674
622	Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin	Diederik P Kingma 和 Jimmy Ba。2014. 亚当:一种随机优化方法。	675
623	王,胡国平。2020. 重温预训练	arXiv 预印本arXiv:1412.6980。	676
624	中文自然语言处理模型。在		677
625	EMNLP 会议记录:调查结果,第 657-668 页。	Thomas N. Kipf 和 Max Welling。2017. 图卷积网络的半监督分类。	678
626	Daniel Dahlmeier 和 Hwee Tou Ng。2012. 更好	在 ICLR 会议记录中。	679
627	评估语法错误纠正。在 NAACL-HLT 会议记录中,第 568-572		680
628	页。	清野俊、铃木润、三田正人、水本智也、犬井健太郎。2019. 将伪数据纳入	681
629	Daniel Dahlmeier,Hwee Tou Ng 和 Siew Mei Wu。	语法纠错的实证研究。在 EMNLP-IJCNLP 会议记录中,第1236-1242	682
630	2013. 构建大型带注释的学习者语料库	页。	683
631	英语:学习者英语的 nus 语料库。在 BEA@NAACL-HLT 会议记录中,		684
632	第 22-31 页。	Mike Lewis,Yinhan Liu,Naman Goyal,Marjan Ghazvininejad、	685
633	马库斯·迪金森和玛尔瓦·拉格布。2009. 学习者语料库的依赖注释。在	Abdelrahman Mohamed,Omer Levy、Veselin Stoyanov 和	686
634	诉讼中	Luke Zettlemoyer。2020. Bart:用于自然语言生成、翻译和理解的	687
635	树库和语言理论国际研讨会,第 59 页。	序列到序列去噪预训练。在 ACL 会议记录中,第 7871-7880 页。	688
636			689
			690
			691

692	李祖超、凯文·帕诺和赵海。 2022. 在语法中结合丰富的语法信息	Marek Rei 和 Helen Yannakoudakis。 2016.学习者写作中错误检	745
693		测的组合序列标记模型。在 ACL 会议记录中,第1181-1191 页。	746
694	纠错。信息处理和管理。		747
695			748
696	Jared Lichtarge,Chris Alberti 和 Shankar Kumar。	Sascha Rothe,Jonathan Mallinson,Eric Malmi,Sebastian	749
697	2020. 用于语法纠错的数据加权训练策略。 TACL,第 634-646	Krause 和 Aliaksei Severyn。 2021.多语言语法错误纠正的简	750
698	页。	单方法。	751
		在 ACL-IJCNLP 会议记录中,第 702-707 页。	752
699	杰瑞德·利希塔奇、克里斯·阿尔贝蒂、尚卡尔·库马尔、诺姆	Devendra Sachan,Yuhao Zhang,Peng Qi 和 William L	753
700	Shazeer,Niki Parmar 和 Simon Tong。 2019.Corpora生成	Hamilton。 2021. 语法树是否有助于预训练的转换器提取信息?	754
701	语法错误纠正。在	在EACL会议记录中,第 2647-2661 页。	755
702	NAACL-HLT 会议记录,第 3291-3301 页。		756
703	三田正人、清野俊、金子正弘、Jun		
704	Suzuki, and Kentaro Inui。 2020. 自我完善	里科·森里奇、巴里·哈多和亚历山德拉·伯奇。	757
705	语法错误校正中的降噪策略。在 EMNLP 会议记录 (调	2016. 带有子词单元的稀有词的神经机器翻译。在 ACL	758
706	查结果)中,页面	会议记录中,第 1715-1725 页。	759
707	267-280。		760
708	永田凉和坂口圭佑。 2016. 学习者英语的短语结构注释和解析。在	Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei	761
709		Yang, Li Zhe, Hujun Bao, and Xipeng Qiu.	762
710	ACL 会议记录,第 1837-1847 页。	2021. Cpt: 一种用于汉语理解和生成的预训练不平衡变压器。	763
		arXiv 预印本 arXiv:2109.05729。	764
711	维诺德奈尔和杰弗里 E 辛顿。 2010. 整改		765
712	线性单元改进了受限的 Boltzmann 机器。		
713	在 ICML 会议记录中,第 807-814 页。	菲利克斯斯塔尔伯格和尚卡尔库马尔。 2021.合成数据生成语法错	766
		误校正与标记的腐败模型。在BEA@EACL会议记录中,第 37-47	767
714	Courtney Napoles,Keisuke Sakaguchi,Matt Post 和	页。	768
715	乔尔·泰特罗。 2015. 语法纠错指标的基本事实。在 ACL 诉讼中		769
716			
717	(短),第 588-593 页。	Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and	770
		Houfeng Wang。 2022. A unified strategy for multilingual	771
718	考特尼·拿破仑、坂口圭介和乔尔	grammatical error correction with pre- trained cross-	772
719	特特罗。 2017. Jfleg:语法错误纠正的流利语料库和基准。在	lingual language model. arXiv preprint arXiv:2201.10707.	773
720	EACL 会议记录中,第 229-234 页。		774
721			
722	特德布里斯科,克里斯蒂安	Christian Szegedy,Vincent Vanhoucke,Sergey Ioffe、 Jon	775
723	哈迪维诺托、雷蒙德·亨迪·苏桑托和克里斯托弗·布莱恩特。	Shlens 和 Zbigniew Wojna。 2016. 重新思考计算机视觉的初	776
724	2014. conll-2014 共享任务	始架构。在ICCV会议记录中,第 2818-2826 页。	777
725	语法错误更正。在诉讼中		778
726	CoNLL:共享任务,第 1-14 页。	Ashish Vaswani,Noam Shazeer,Niki Parmar,Jacob	779
		Uszkoreit,Llion Jones,Aidan N Gomez、 ukasz	780
727	Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem	Kaiser 和 Illia Polosukhin。 2017. 注意力就是你所需	781
728	Chernodub 和 Oleksandr Skurzhashkyi。 2020 年。	要的。在 NIPS 会议记录中,第 5998-6008 页。	782
729	向量语法纠错:标记,不是		
730	改写。在 BEA@ACL 会议记录中,第 163 页-	Petar Velickovic,Guillem Cucurull,Arantxa Casanova、	783
731	170。	Adriana Romero,Pietro Liò 和 Yoshua Bengio。	784
		2018. 图形注意网络。在ICLR会议记录中。	785
732	迈尔·奥特、谢尔盖·埃杜诺夫、阿列克谢·巴耶夫斯基、安吉拉		786
733	Fan,Sam Gross,Nathan Ng,David Grangier 和		
734	迈克尔·奥利。 2019. fairseq:一个快速、可扩展的	万兆宏和万小军。 2021. 具有依赖树校正的语法引导语法纠错模	787
735	序列建模工具包。在诉讼中	型。 arXiv 预印本arXiv:2111.03294。	788
736	NAACL-HLT (演示),第 48-53 页。		789
			790
737	斯拉夫彼得罗夫和瑞恩麦克唐纳。 2012. 概述		
738	2012 年关于解析网络的共享任务。	王宇、王岳林、党凯、刘杰、刘卓。 2021.语法错误纠正的综合调查。	791
		ACM 智能系统和技术交易 (TIST),第 1-51 页。	792
739	科林·拉菲尔、诺姆·沙泽尔、亚当·罗伯茨、凯瑟琳		793
740	Lee, Sharan Narang, Michael Matena, Yanqi Zhou,		794
741	李伟和彼得 J 刘。 2020.用统一的文本到文本探索迁移学习的极		
742	限	罗纳德·J·威廉姆斯和大卫·齐泽。 1989. 一种持续运行完全循环神	795
743	变压器。机器学习研究杂志	经网络的学习算法。神经计算,1 (2) :270-280。	796
744	(JMLR), 21(140):1-67。		797

798	山姆怀斯曼和亚历山大 M. 拉什。 2016. 序列到序列学习作为波	赵媛媛、姜南、孙薇薇、万晓军。 2018. nlpcc 2018 共享任务概	851
799	束搜索优化。在	述:语法纠错。在 CCF自然语言处理和中文计算国际会议	852
800	EMNLP 会议记录,第 1296-1306 页。	(NLPCC),第 439-445 页。	853
801	托马斯·沃尔夫,Lysandre Debut,维克多·桑,朱利安		854
802	肖蒙德·克莱门特·德朗格、安东尼·莫伊·皮埃里克·西斯塔克、蒂		855
803	姆·劳特·雷米·卢夫、摩根·冯托维奇、		
804	乔·戴维森、山姆·施莱弗、帕特里克·冯·普拉滕、克拉拉		
805	Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le		
806	Scoo, Sylvain Gugger, Mariama Drame, 昆汀		
807	Lhoest 和 Alexander M. Rush。 2020.变形金刚:最先进的		
808	自然语言处理。在		
809	EMNLP 会议记录(演示),第 38-45 页。		
810	Qingrong Xia, Zhenghua Li, Min Zhang, Meishan		
811	Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019.		
812	语法感知神经语义角色标签。在AAAI 会议记录中,第		
813	7305-7313 页。		
814	Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. A		
815	联合汉语分词和依存分析的基于图的模型。 TACL,第 78-92		
816	页。		
817	海伦·扬纳库达基斯、泰德·布里斯科和本·梅德洛克。		
818	2011. 一种新的自动数据集和方法		
819	分级esol文本。在 ACL 会议记录中,页面		
820	180-189。		
821	Michihiro Yasunaga、Jure Leskovec 和 Percy Liang。		
822	2021.lm-critic:无监督语法错误纠正的语言模型。在		
823	诉讼中		
824	EMNLP,第 7752-7763 页。		
825	郑源和特德布里斯科。 2016. 使用神经机器翻译进行语法错误		
826	校正。在		
827	NAACL-HLT 会议记录,第 380-386 页。		
828	郑源、Shiva Taslimipoor、Christopher Davis 和		
829	克里斯托弗·布莱恩特。 2021.多类语法		
830	用于纠正的错误检测:两个系统的故事。		
831	在 EMNLP 会议记录中,第 8722-8736 页。		
832	张宝林。 2009. HSK的特点和功能		
833	动态作文语料库。国际华人		
834	语言教育,4:71-79。		
835	Bo Zhang, Yue Zhang, Rui Wang, Zhenghua Li, and		
836	张敏。 2020a。具有依赖图卷积网络的语法感知意见		
837	角色标签。		
838	在 ACL 会议记录中,第 3249-3258 页。		
839	Meishan Zhang, Zhenghua Li, Guohong Fu, and Min		
840	张。 2019.具有句法感知词表示的句法增强神经机器翻译。在		
841			
842	NAACL-HLT 会议记录,第 1151-1161 页。		
843	张宇、李政华、张敏。 2020b。用于神经依赖的有效二阶 treecrf		
844			
845	解析。在 ACL 会议记录中,第 3295-3305 页。		
846	Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li,		
847	Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022.		
848	MuCGEC:多参考多源评估		
849	用于中文语法纠错的数据集。在		
850	NAACL-HLT 会议记录。		

BEA-19-dev FCE-测试			
动车组	F0.5	F0.5	
贝尔等人。(2019)	FCE-train	48.50	57.28
Kaneko and Komachi (2019)	FCE-train	-	61.65
al. (2021)	FCE-train	65.54	72.93
GOPar FCE-train	GOPar CLang8	66.32	74.10
		72.13	71.53

表 6:二进制 GED 性能。

超参数

我们实验中使用的主要超参数如表7 所示。不使用PLM 时，总训练时间约为 3 小时。

使用 PLM 时,培训大约需要 7 个小时。

B PLM 的下载链接

下面列出了我们实验中使用的 PLM 的下载链接。我们雇用伊莱克特拉 (Clark 等人, 2020 年; Cui 等人, 2020 年)构建 GOPar 和 BART (Lewis 等人, 2020 年; Shao 等人, 2021 年)增强我们的 GEC 模型。

- ELECTRA-English-Large。
- ELECTRA-中文-大。
- BART-English-Large。
- BART-中文-大。

配置	价值
预训练	
基础架构	变压器底座 (不带 PLM) 大型变压器 (带 PLM)
预训练语言模型	BART-large (Lewis 等人, 2020)
时期数	60
设备	8 个特斯拉 V100 GPU (32GB)
每个 GPU 的批量大小	8096 个令牌
优化器	亚当(Kingma and Ba, 2014)
学习率	(β1 “0.9, β2 “0.98, ε “1 ^ 10 ^ -8)
热身更新	5 ^ 10 ^ -4
最大源长度	4000
DepGCN 层数	64 (英文) ; 128 (中文)
双上下文聚合 β	3
	0.5
损失函数	标签平滑交叉熵 (标签平滑=0.1)
退出	(Szegedy 等人, 2016 年)
辍学-src	0.1 (无 PLM) ; 0.3 (带 PLM)
	0.2
微调	
学习率	5 ^ 10 ^ -5
热身更新	1000
一代	
光束尺寸	12
最大输入长度	64 (英文) ; 128 (中文)

表 7:我们实验中使用的超参数值。

PLM 语法GLUE			
基线	58.15 SynGEC	√ 60.14 +1.99	-
基线/ SynGEC	√	61.53	-
		√ 62.15 +0.62	

表 8:不同模型在 JF LEG 基准上的 GLUE 分数。

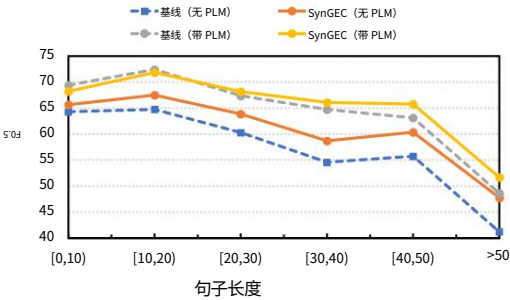


图 5:输入句子长度的影响。

C GOPar 的 GED 能力

我们评估了 GOPar 在两个主流 GED 数据集上的二进制语法错误检测 (GED) 性能,即 BEA-19-dev (Bryant et al., 2019)和 FCE-test (Yannakoudakis et al., 2011) 我们遵循Rei 和 Yannakoudakis (2016)并报告令牌级 P/R/F 值以检测不正确的标签。表6显示了GOPar 和其他领先的 GED 模型的性能。在使用相同的训练数据时,GOPar 具有超强的语法错误检测能力。这种现象非常有趣,值得更深入的研究。

D JFLEG 实验

JFLEG (Napoles et al., 2017)是一个英语 GEC 评估数据集,专注于流畅度,使用 GLUE 分数(Napoles et al., 2015)作为评估指标。我们在 JFLEG 上评估表2中的基线和SynGEC 方法。由于JFLEG 的规模相对较小 (只有 747 句),我们选择在附录中呈现结果。从表8 中,我们可以看到,句法知识仍然在有/没有 PLM 的基线上不断提高 GEC 性能。

E 关于输入的性能
句子长度

图5显示了 CoNLL-14 测试在输入长度方面的性能。当遇到长句子时,基线会严重退化。

900 引入句法知识后就清楚了
901 在更长的时间内获得更大的改进
902 两种情况下的句子有或没有
903 PLM,因为语法有助于解决与
904 远程信息,这在长篇中很常见
905 句子。