基于字词粒度噪声数据增强的中文语法纠错

汤泽成[1]　纪一心[1]　赵怡博[2]　李军辉[1]
（苏州大学计算机科学与技术学院，苏州，江苏，215006）
[1]{zctang2000，jiyixin169，lijunhui26}@gmail.com
[2]1815406006@stu.suda.edu.cn

摘要

语法纠错是自然语言处理领域的热门任务之一，其目的是将错误的句子修改为正确的句子。为了缓解中文训练语料不足的问题，本文从数据增强的角度出发，提出一种新颖的扩充和增强数据的方法。具体地，为了使模型能更好地获取不同类型和不同粒度的错误，本文首先对语法纠错中出现的错误进行了字和词粒度的分类，在此基础上提出了融合字词粒度噪声的数据增强方法，以此获得大规模且质量较高的错误数据集。基于NLPCC2018共享任务的实验结果表明，本文提出的融合字词粒度加噪方法能够显著提升模型的性能，在该数据集上达到了最优的性能。最后，本文分析了错误类型和数据规模对中文语法纠错模型性能的影响。

关键词： 中文语法纠错；字和词粒度；数据增强；多头注意力

Chinese Grammatical Error Correction enhanced by Data Augmentation from Word and Character Levels

Zecheng Tang[1]　Yixin Ji[1]　Yibo Zhao[2]　Junhui Li[1]
(School of Computer Science and Technology，Soochow University，
Suzhou，Jiangsu，215006)
[1]{zctang2000，jiyixin169，lijunhui26}@gmail.com
[2]1815406006@stu.suda.edu.cn

Abstract

Grammatical error correction (GEC) is one of the popular tasks in natural language processing, and it aims to transfer a sentence with errors into a correct one. To alleviate the issue of limited annotated corpora of Chinese GEC, in this paper we propose a novel data augmentation method to enhance the training dataset. Specifically, we first carefully categorise errors in Chinese GEC into multiple groups from both word and character granularities. Experimental results on NLLPCC 2018 shared task show that by adding noises from both word and character granularities significantly improves the performance of Chinese EGC, and achieves a new state-of-the-art. Moreover, we analysis the effect of different error types and scale of external dataset on the performance of Chinese GEC.

Keywords: Chinese grammatical error correction , character and word granularities , data augmentation , multi-headed attention

基金项目：苏州大学‘大学生创新创业训练计划’基金资助(202010285095Y)

1 引言

语法纠错(grammatical error correction, GEC)是自然语言处理领域的重要任务之一，语法纠错的目标是给定一个包含语法错误的句子，通过模型将其修正成正确的句子。许多研究将语法纠错当成一个机器翻译(machine translation，MT)任务，其源端是一个错误的句子，目标端是一个正确的句子(Yuan and Briscoe, 2016)。例如，给定一个带有语法错误的句子：小明**金**天去上学，通过纠正错误得到其对应正确的句子：小明**今**天去上学。

相比于英文，中文语法纠错任务的难点在于，中文的词法要比英文复杂得多。在英文中，词之间是以空格作为自然分界符的；而在中文中，最小的有意义的单位是词，但词没有一个形式上的分界符。例如，如果将词语“明天”简单分为“明”和“天”，那么这个词就失去了原本的语义。这样，引起中文语法错误的既可能是词，也可能是词内部的某个字。同时，由于中文语法纠错起步时间相对较晚，人工标注的语料规模仍然非常有限。因此，如何通过有限的语料，让模型较好地自动纠正中文语法中的错误已成为目前研究的热点。

为了弥补了上述数据缺少的问题，很多研究采用数据增强的方式扩充数据集(Zhao et al., 2019; Xie et al., 2018; Rei et al., 2017)。针对中文语法纠错，现有工作(王辰成et al., 2020)以词为单位，通过加噪的方式以扩充数据集。然而，考虑到中文词法的复杂性，仅仅以词为单位生成的含噪语料无法覆盖很多真实的错误源：词的内部发生了错误。例如，对于句子“小明今天去上学”，如果以词为单位进行加噪，将不能够得到“小明令天去上学”这种在词内部存在错误的句子。因此，为了弥补仅以词为单位进行数据扩展存在的覆盖率问题，本文将对中文句子同时以词和以字为单位进行数据增强，从而使得扩充后的数据更符合真实情况。

具体地，本文将通过噪声生成和预训练结合的方法提升中文语法纠错的性能。在模型的选取方面，本文使用基于多头注意力的Transformer序列到序列模型(Vaswani et al., 2017)，同时引入源端词Dropout的方法，以减少模型对源端词的依赖性。基于NLPCC2018中文语法纠错数据集的实验结果表明，本文提出的融合字词粒度噪声的数据扩展的方法（即以词和字为单位同时进行），显著提升了中文语法纠错的性能。同时，本文的模型在该数据集上获取了目前最优的性能。

本文的贡献在于：

- 针对中文语法纠错任务，首次提出同时融合字词粒度噪声的数据扩展方法；
- 基于NLPCC2018公开评测数据，在仅使用原始Transformer模型的情况下，本文获取了目前最优的性能。

2 相关工作

早期，由Yu(2014)等人发起的中文语法错误诊断(Chinese Grammatical Error Diagnosis, CGED)测评任务中，包含中文语法中常见的四种错误类型：冗余、缺失、语序和词语选择。针对该问题，Lee(2014)等人使用n-gram统计特征并制定规则，提出了句子级别的语法诊断系统，该方法在规则的基础上对每条输入的训练句子打分，从而判断句子的正确性。Yu(2014)等人则提出以字符级别的n-gram语言模型找出可疑字，通过邻近字组合和字典查询的方式找出概率最大的候选字。以上的方法均属于人工提取特征，这些特征并不完整，有可能丢失一些重要信息。为了解决这个问题，Zheng(2016)等人提出了三个纠错模型，分别是基于CRF的模型、基于LSTM的模型和使用堆叠的模型，相比于人为地抽取特征，基于LSTM的神经网络方法可以自动学习并抽取出语言的特征。进一步的，Xie(2017)等人将LSTM模型与CRF模型进行融合，可以更有效地预测出标记序列。

Yuan和Briscoe(2016)首次将语法纠错任务看作是一个序列到序列任务，使用机器翻译模型取得了较好的性能。之后，序列到序列模型被广泛应用于语法纠错任务中。在英文语法纠错任务中，Chollampatt等(2018)采用多层卷积网络捕捉长距离信息，从而使神经机器翻译模型在语法纠错任务上第一次超过统计机器翻译。Grundkewicz等(2018)将统计机器翻译和神经机器翻译结合，达到了很好的纠错效果。而在中文语法纠错任务中，Ren(2018)等人将中文语法纠错当成翻译任务，提出一种基于卷积神经网络的纠错模型，在纠错的准确率方面取得了最好的效果。Duan(2021)等人在Transformer模型基础上进行改进，进一步提高模型在局部语句上的纠错能力。

虽然在模型上有了很大的改进，但是学者逐渐意识到语料集的规模和错误种类的覆盖率对模型效果的提升起到关键作用。Koehn和Knowles(2017)发现基于编码器和解码器的神经机器翻译需要大量的语料，数据规模和质量对模型性能有很大的影响。为此，Xie等(2018)首次提出了使用伪语料进行数据增强的方式，王辰成等(2020)提出一种基于规则的腐化语料集方法增加错误数据规模。Zhao等(2020)则提出一种动态随机掩盖的方式，通过掩盖并替换部分数据达到扩充语料集的目的。上述的几种方式均取得了不错的效果。

随着CCF国际自然语言处理和中文计算会议(NLPCC)在2018年首次增加了中文语法错误纠正任务(Zhao et al., 2018)，更多的学者投入到中文语法纠错的研究中。Youdao团队(Fu et al., 2018)将错误划分为表面错误和语法错误，使用近音字和5-gram语言模型解决表面错误，再用Transformer模型解决高级错误，对纠正后的句子进行困惑度分析，选择最优的句子作为输出的结果。阿里团队(Zhou et al., 2018)则使用了多种模型，将错误划分，通过对类别进行候选和组合，得到最终的纠正结果。

3 基于源端词Dropout的Transformer模型

Transformer是一种基于多头注意力的序列到序列模型，它由一个编码器和一个解码器构成。编码器将输入编码为高维隐含语义向量，并将结果输入到解码器中，解码器通过自回归的方式生成最终预测的结果。编码器包含多个编码模块，解码器则包含多个解码模块。在模型源端的输入部分，使用源端词Dropout方法随机丢弃部分词向量，总体模型架构如图1所示。下面将分别介绍编码器、解码器和源端词Dropout方法。

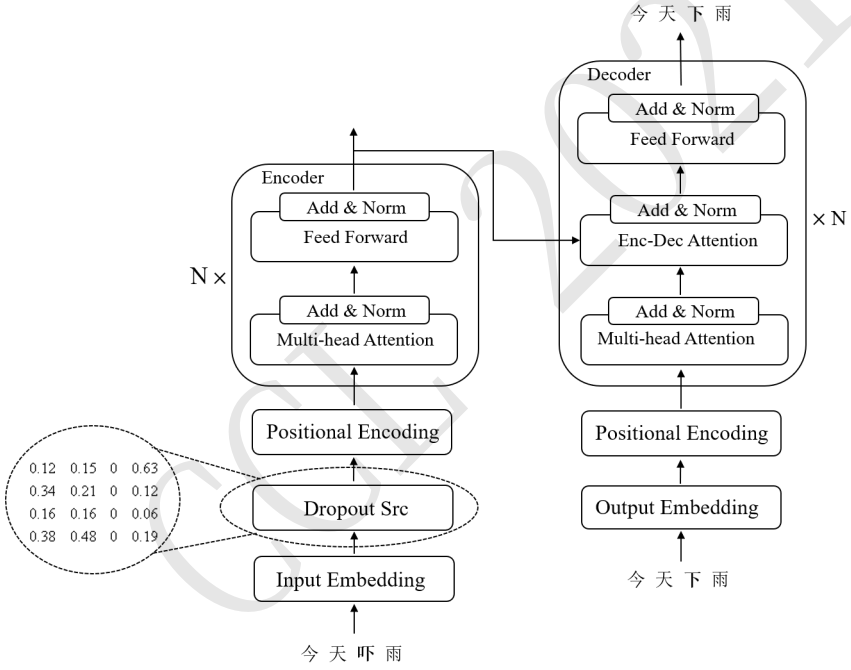

图 1: 引入源端词Dropout的Transformer模型架构

3.1 编码器

编码器由N个完全一样的编码模块拼接而成的，每个模块包含三个部分，分别是多头自注意力层、前馈传播层以及残差连接层三个子层。多头自注意力层可以建立输入序列间的长距离依赖关系，其公式如式(1)(2)(3)所示：

$$\text{Attention}\left(Q,K,V\right)=\text{Softmax}\left(\frac{QK^{T}}{\sqrt{d_{k}}}\right)V \tag{1}$$

$$head_{i}=\text{Attention}\left(Q_{i},K_{i},V_{i}\right) \tag{2}$$

$$\text{MultiHead}\left(Q,K,V\right)=\text{Concat}\left(head_1,head_2\ldots,head_n\right)W^o \tag{3}$$

其中Q、K和V分别查询矩阵(query)、键值矩阵（key)和实值矩阵(value)，它们是由输入向量通过三个不同的线性层得到的。d_k表示词嵌入层的维度，该因子是为了防止在前馈和反馈传播的过程中因为内积过大造成的概率分布不均匀。每个注意力层(Attention)都是采用点积的方式进行计算。为了满足多头注意力的特征，模型分别在不同的子空间内计算各自的注意力矩阵，最后将它们拼接起来，以此获得不同子空间内的相关信息。下标i表示对应第i个自注意力矩阵。

残差连接层(He et al., 2016)位于每两个子层之间，对上一子层的输出进行归一化(Ba et al., 2016)并与上一子层的输入进行累加，以防止出现反向传播时梯度消失的问题，如式4所示。

$$y=\text{LayerNorm}\left(F\left(x\right)+x\right) \tag{4}$$

其中x为上一个子层网络的输入，F(·)和LayerNorm(·)分别表示上一个子层网络和归一化函数。

前馈传播层是由两个线性层构成，它们拥有独立的权重、偏置以及维度，可以进一步提取语义信息。

3.2 解码器

解码器是由N个完全一样的解码模块构成，除了与编码模块中完全一样的两个部分以外，还包含了一个编码-解码自注意力层(Enc-Dec Attention)。该层的Q(查询矩阵)和K(键值矩阵)均为编码器的输出向量，而V(实值矩阵)是解码器多头自注意力层的输出。此层的目的是为了将源端和目标端的输入特征进行融合，即计算编码器和解码器向量间的注意力。为了保持自回归的特性，解码器依赖上一步的预测输出序列，每次输出一个词，最后遇到句子末尾的特殊标记符结束，将所有单词拼接在一起获得最后的输出。

3.3 源端词Dropout

受到Junczys-Dowmunt等人(2018)工作的启发，为解决错误数据规模不足的问题，一方面可以对数据加入人工噪声，另一方面可以减少编码器对源端错误信息的依赖性。和传统Dropout(Srivastava et al., 2014)不同，源端词Dropout在源端数据经过词嵌入层后随机将一部分词向量置为0。基于此，在源端以P_{src}的概率筛选出要被丢弃的单词，将它们的词向量设置为0，同时，要对未被丢弃的其他词向量做出相应的补偿，以$1/(1-P_{src})$的倍率进行放缩。通过这种方法，模型没有得到完整的源端输入，它的泛化能力更强，可以更积极地在数据上进行纠错。

4 融合字词粒度噪声的数据增强方法

提升模型性能最简单有效的方式是提升数据规模和提高数据质量。在中文语法纠错领域，为了缓解人工标注语料库规模受限的问题，本章将首先对语法纠错中出现的错误进行分类（第4.1节），然后再使用人工加噪的方式提升数据的规模（第4.2节）。

4.1 错误类型的字词粒度划分

在传统直接生成噪声语料(Edunov et al., 2018; Zhao et al., 2019)的过程中，仅仅是对语料加上词粒度的噪声。例如，对于“今天出太阳了，天气很好”这句话中的“天气”一词，如果按照传统加噪方式，可能会将“天气”替换为“树木”，于是加噪后的句子变为“今天出太阳了，树木很好”。但是，这种加噪方式无法应用在字粒度上——例如将“天气”中的“气”字变为“汽”。如果将字粒度和词粒度结合起来，可以使模型学习到更加完备的错误类型以及错误类型间的不同组合。受到Rao等人(2017)工作的启发，我们在现有错误类型上划分了4个大类和13个子类，如表1所示，其中加粗为错误部分。

4.2 融合字词粒度噪声的数据增强

为了融合不同粒度的错误类型，本文提出了一种融合字词粒度噪声的数据增强方法，同时提出一种复制累加机制提升数据规模。如图(2)所示，首先将源端数据进行复制，对前四份分别加上不同类型的错误，对最后一份数据加上所有类型的错误。针对每份语料，首先进行词粒度

原句	我希望您尽快把问题解决。		
冗余(redundant)	词粒度	词冗余	我 希望 **期望** 您 尽快 把 问题 解决。
	字粒度	字冗余	我 希 望 您 尽 **能** 快 把 问 题 解 决 。
缺词(missing)	词粒度	缺词	我 希望 您 尽快 把 解决 。
	字粒度	缺字	我 希 望 您 尽 快 把 问 解 决 。
选词(selection)	词粒度	词错误	我 希望 您 **确实** 把 问题 解决。
	字粒度	同音字	我 希 **忘** 您 尽 快 把 问 题 解 决 。
		形近字	我 希 望 您 尽 快 把 **间** 题 解 决 。
		其他字错误	我 希 望 您 **数** 快 把 问 题 解 决。
词序(ordering)	词粒度	词间语序	我 希望 您 **解决** 把 问题 **尽快**。
	字粒度	词内语序	我 希 望 您 **快 尽** 把 问 题 解 决 。

表 1: 基于字词粒度的错误类型划分

噪声化，然后进行字粒度噪声化，将这两个数据增强步骤统称为双重噪声化。最后，将这五份噪声化后的语料进行叠加，得到最终的加噪语料。冗余错误类型的噪声化实现如算法(1)所示，其余三种错误类型(缺词、词序、选词)的数据增强方式只需要将第10、11、23、24行按规则进行修改即可。其中第2∼14行对应词粒度的加噪，第15∼27行对应字粒度的加噪，对于综合噪声化，只需要在第10、11、23、24行通过随机数决策的方式决定对这个单词或字增加任意一种噪声即可。

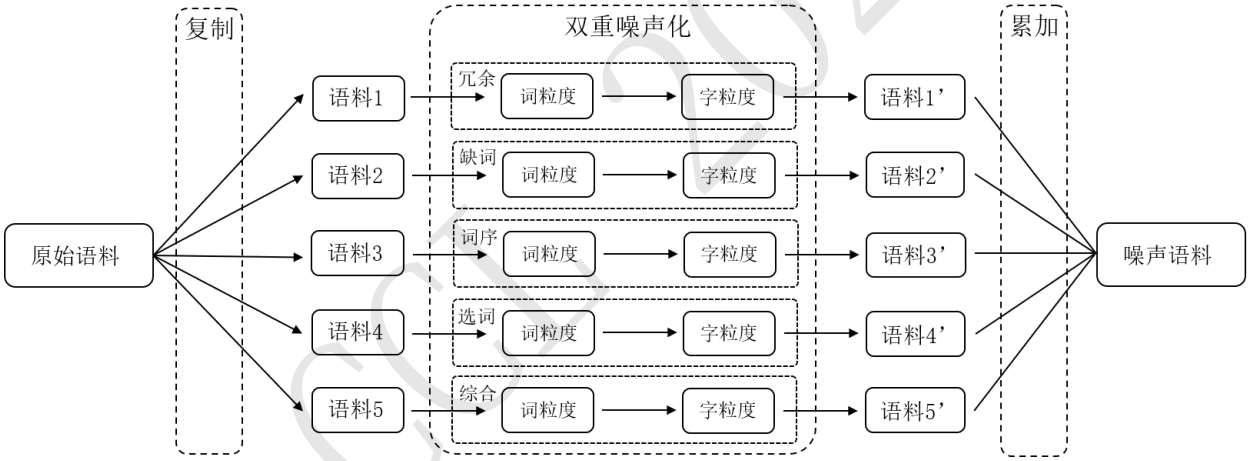

图 2: 复制累加噪声化

根据训练数据的最短编辑距离，可以得到整份语料的错误率在30%左右，为了使双重噪声化后语料的错误率也维持在30%，利用数学期望计算（推导过程见附录A），得到P_{rate}为0.163。针对每份复制语料，根据算法(1)加上了不同类型的噪声，最后再将所有的噪声语料进行累加构成源端并生成平行数据，最终数据增强的效果如表(2)所示。

通过这种复制累加的方式，可以高效地使用少量正确数据生成大规模的覆盖不同错误类型的数据。虽然在噪声化后的语料已经完全不符合中文的语言规律，但是神经网络可以通过抽取高维的语义特征使模型学习到相应的错误，我们将在第5.4节验证这种方式的有效性。

5 实验

基于上述提出的方法，对实验过程做出了如下的规定：1)将Gigaword[0]数据集作为外部语料，NLPCC2018官方提供的数据集作为内部语料；2)对于外部噪声化语料，采用预训练的方

[0]https://catalog.ldc.upenn.edu/LDC2009T27

Algorithm 1 融合字词粒度噪声的数据增强算法（冗余）

Input: 复制语料C_m，每个单词/字被修改的概率P_{rate}
Output: 噪声语料C_{noise}

begin
加载语料，对语料进行分词
统计分词后数据包含的单词，构建词表D_1
for S in C_m **do** /*遍历语料C_m中的每个句子S*/
  for W in S **do** /*遍历句子S中的单词W*/
    获取一个随机数K
    if $K \geq \mathrm{P}_{rate}$ **then**
      Continue
    end if
    从D_1中随机取一个单词W_1
    在W左侧添加W_1
    将加完噪声的句子放到临时语料C_{tmp}中
  end for
end for
对C_{tmp}进行分字
统计分字后数据包含的字数，构建词表D_2
for S in C_{tmp} **do** /*遍历语料C_{tmp}中的每个句子S*/
  for W in S **do** /*遍历句子S中的字W*/
    获取一个随机数K
    if $K \geq \mathrm{P}_{rate}$ **then**
      Continue
    end if
    从D_2中随机取一个字W_2
    在W左侧添加W_2
    将加完噪声的句子放到临时语料C_{noise}中
  end for
end for
输出噪声语料C_{noise}

式训练一个seq2seq(Sutskever et al., 2014)模型；3)用内部语料在预训练模型上进行微调；4)所有语料在分词后进行加噪，防止因为分词导致字字粒度和词粒度的加噪混淆；5)所有实验使用Fairseq开源代码库[1]。下面是实验的设置。

5.1 实验数据及预处理

本文采用的平行语料包括NLPCC2018官方提供的Lang-8语料集[2]以及HSK汉语考试语料集(张宝林, 2008)，首先对内部语料集进行处理生成平行语料，得到1,377,776对训练语句，然后从中抽出5,000条平行语料生成开发集。为了与前人的工作进行对比(见5.4节)，统一将外部数据规模设置为600万。由于采用了复制累加机制，所以对于本文提出的方法，只需从Gigaword中抽取出120万条原始复制语料，经过累加机制后可以自动扩充为600万。在所有抽取出的语料中，长度超过35的句子数目和未超过35的句子数目比例为3:2。所有语料的规模如表(3)所示。在进行双重噪声化的过程中，首先采用jieba分词工具[3]进行分词并加上噪声，然后再进行分字操作并加上噪声得到最终的含噪预训练语料。由于官方提供的测评数据使用pkunlp工具包[4]进行分词，所以在得到结果后，先将结果还原成未分词的状态，然后使用pkunlp工具包进行分词得到最后的结果。

[1]https://github.com/pytorch/fairseq
[2]http://tcci.ccf.org.cn/conference/2018/dldoc/trainingdata02.tar.gz
[3]https://github.com/fxsjy/jieba
[4]http://59.108.48.12/lcwm/pkunlp/downloads/libgrass-ui.tar.gz

原始数据	请你告诉我他的名字，我想不起来了。
Char分词	请 你 告 诉 我 他 的 名 字 ， 我 想 不 起 来 了 。
冗余噪声化	请 你 他 告 诉 我 作 者 他 的 名 字 天 刊 ， 我 想 不 雨 伞 起 来 了 。
缺词噪声化	请 告 诉 他 的 ， 我 想 不 起 。
词序噪声化	告 诉 请 你 我 的 名 字 ， 我 他 想 不 来 了 起 。
选词噪声化	请 湖 泊 告 诉 电 他 的 错 字 ， 我 想 很 起 水 了 。
综合噪声化	请 你 物 探 诉 你 他 的 名 坍 ， 我 这 里 起 来 新 了 。

表 2: 数据增强效果示例

语料名称	句子数	单词数
Lang-8	1,220,906	23,707,326
HSK	156,870	4,252,643
Gigaword原始复制语料	1,200,000	40,613,923
Gigaword对比实验语料	6,000,000	319,102,506
NLPCC2018测试集	2,000	58,768

表 3: 数据集规模统计

5.2 模型参数设置

使用Transformer模型作为基本模型，模型的超参数为：源端和目标端词嵌入矩阵维度均为512，并且共享词嵌入矩阵的权重。编码器和解码器均包含6个神经模块，每个多头自注意力层均包含8个头，前馈层的第二个维度为2048。Dropout的概率设置为0.3，P_{src}设置为0.3。优化器采用Adam(Kingma and Ba, 2014)，学习率初始值为1×10^{-7}，在前4000轮中逐渐线性增长到5×10^{-4}，接着逐渐下降。在每轮训练结束时会在开发集上进行一次验证，通过观察在开发集上的损失值判断模型是否收敛，直到损失函数在10轮内相差不超过1×10^{-5}时训练结束。在最终通过测试集生成数据的时候，将柱状搜索的大小设置为12，批大小设置为1024个句子，随机使用5次不同的种子，将最后的分数取平均作为最终的结果。

5.3 评价指标

采用公开的MaxMatch(M2)工具包[5]计算$F_{0.5}$，其计算的公式如式(5)～(7)所示。

$$P = \frac{\sum_{i=1}^{N} |e_i \cap ge_i|}{\sum_{i=1}^{N} |e_i|} \tag{5}$$

$$R = \frac{\sum_{i=1}^{N} |e_i \cap ge_i|}{\sum_{i=1}^{N} |g_i|} \tag{6}$$

$$F_{0.5} = \frac{(1 + 0.5^2) \times R \times P}{0.5^2 \times P + R} \tag{7}$$

其中，P (Precision)表示准确率，R (Recall)表示召回率，$F_{0.5}$是最终的得分，N是句子总数，e_i是模型对第i个错误句子修改的集合，ge_i是对第i个错误句子的标准修改集合，$|e_i \cap ge_i|$表示模型对句子i修改与标准修改匹配的数目。

[5]https://github.com/nusnlp/m2scorer

5.4 实验结果

采用上述$F_{0.5}$作为测评标准，char-Transformer作为基准模型，表(4)展示了实验的结果。在char分词的基础上使用源端词Dropout的方法后，模型的性能有进一步的提升。为了在数据增强方式上进行对比，本文使用了王辰成等(2020)提出的腐化语料集方法，在保证同样的开源工具库、超参数、模型和数据规模的情况下进行实验。可以发现，在数据规模扩充后，模型的性能有了很大提升，这符合Koehn和Knowles提出的观点(Koehn and Knowles, 2017)，同时也证明了融合字词粒度噪声的数据增强方式可以使模型学到更多类型的错误，所以在性能上有所提升。

模型	P	R	$F_{0.5}$	$\Delta F_{0.5}$
char-Trans	41.24	17.17	32.21	-
jieba-Trans	35.68	20.16	30.92	- 1.29
char-Trans + 源端词Dropout	43.93	20.21	35.58	+ 3.37
char-Trans + 源端词Dropout + 腐化语料集	45.70	22.94	38.13	+ 5.91
char-Trans + 源端词Dropout + 融合字词粒度噪声	**47.29**	**23.89**	**39.49**	+ 7.28

表 4: 不同改进对模型性能的影响

通过与其他NLPCC2018中文语法纠错任务模型进行性能上的对比，本文提出的单模型性能要优于之前的工作。如表(5)所示，其中Ensemble表示多模型，Single表示单模型。

模型	P	R	$F_{0.5}$
YouDao(Ensemble)(Fu et al., 2018)	35.24	18.64	29.91
AliGM(Ensemble)(Zhou et al., 2018)	41.00	13.75	29.36
BLCU(ensemble)(Ren et al., 2018)	**47.63**	12.56	30.57
C-Transformer(Single)(汪权彬and 谭营, 2020)	38.22	23.72	34.05
动态残差Transformer(Single)(王辰成et al., 2020)	39.43	22.80	34.41
MaskGEC(Single)(Zhao and Wang, 2020)	44.36	22.18	36.97
Ours(Single)	47.29	**23.89**	**39.49**

表 5: 与已有模型性能的对比

6 分析与讨论

6.1 加噪方式对模型性能的影响

为了分析双重噪声化的有效性，我们在相同的数据规模下对两份语料进行了单次噪声化，并将结果与双重噪化话的结果进行比较。对于第一份语料，仅在字级别上进行噪声化；对于第二份语料，仅在词级别上进行噪声化。两次实验所有的超参数设置相同，得到的结果如表(6)所示。实验结果表明，在单次噪声化的情况下，模型的性能都低于使用双重噪声化，但是字级别的噪声化相比于对词级别的噪声化，可以获得更好的性能。

6.2 预训练数据规模对模型性能的影响

为了分析预训练数据规模对模型性能的影响程度，本文分别用不同规模的预训练语料进行对比实验，在其余超参数设置均相同的情况下，实验结果如表(7)所示。实验结果表明，随着数据规模的增长，预训练对模型的性能增益逐渐减小。

加噪方式	P	R	$F_{0.5}$	$\Delta F_{0.5}$
字&词	47.29	**23.89**	**39.49**	-
字	47.15	23.26	39.12	- 0.37
词	**47.51**	22.85	39.07	- 0.42

表 6: 加噪方式对模型性能的影响

数据规模	P	R	$F_{0.5}$	$\Delta F_{0.5}$
600万	47.29	**23.89**	39.49	-
300万	46.66	23.63	39.05	- 0.44
450万	47.06	23.62	39.26	- 0.23
750万	47.31	23.86	39.49	+ 0.00
900万	**47.41**	23.72	**39.51**	+ 0.02

表 7: 数据规模对模型性能的影响

6.3 不同错误类型对模型性能的影响

为了分析不同错误类型对语法纠错模型性能的影响，我们做了四次对比实验，每次实验去除一种错误类型的加噪方式，实验结果如表(8)所示。实验表明每种错误类型的丢失均对最终结果有一定影响。相对而言，语序错误丢失的影响相对较小，而冗余错误和缺词错误的丢失对最后模型性能影响较大。考虑到在测试数据中，所有的错误类型分布趋于均衡，通过对比分析，可以认为冗余和缺词是当前中文上比较常见的语法错误。

模型	P	R	$F_{0.5}$	$\Delta F_{0.5}$
Ours	**47.29**	**23.89**	**39.49**	-
Ours+w/o冗余	46.34	21.58	37.63	- 1.86
Ours+w/o缺词	46.34	21.58	37.69	- 1.80
Ours+w/o选词	45.68	22.56	37.92	- 1.57
Ours+w/o语序	46.12	23.03	38.41	- 1.08

表 8: 不同错误类型对模型性能的影响

7 总结

本文提出一种基于字词粒度噪声的数据增强方法，并使用复制累加机制对数据进行扩充。基于字词粒度的加噪方式可以对数据加上更细粒度的噪声，使模型更好地学习到不同类型的语法错误。复制累加机制可以将原始小规模语料进行数据扩充，有效弥补了语法纠错任务中数据不足的问题。在NLPCC2018测试数据上的实验结果表明，这两种方式的结合可以有效提高语法纠错模型的性能。如何使模型在字词粒度的基础上更好地习得错误类型，我们认为这离不开对语义的理解，因此在理解语义的基础上进行语法纠错是一个值得不断探索的过程。我们在未来的工作中将继续对这一问题进行不断地尝试。

参考文献

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. corr, vol. abs/1607.06450 (2016). *arXiv preprint arxiv:1607.06450*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jianyong Duan, Yang Yuan, and Hao Wang. 2021. Chinese spelling correction method based on transformer local information and syntax enhancement architecture. *Beijing Da Xue Xue Bao*, 57(1):61–67.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

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. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pages 341–350.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. *arXiv preprint arXiv:1804.05945*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. page 28.

Lung-Hao Lee, Liang-Chih Yu, Kuei-Ching Lee, Yuen-Hsien Tseng, Li-Ping Chang, and Hsin-Hsi Chen. 2014. A sentence judgment system for grammatical error detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 67–70.

Gaoqi Rao, Baolin Zhang, Endong Xun, and Lung-Hao Lee. 2017. Ijcnlp-2017 task 1: Chinese grammatical error diagnosis. In *Proceedings of the 2017 International Joint Conference on Natural Language Processing, Shared Tasks*, pages 1–8.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. *arXiv preprint arXiv:1707.05236*.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for chinese grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 401–410.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 619–628.

Yi Yang, Pengjun Xie, Jun Tao, Guangwei Xu, Linlin Li, and Luo Si. 2017. Alibaba at ijcnlp-2017 task 1: Embedding grammatical features into lstms for chinese grammatical error diagnosis task. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 41–46.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 220–223.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning chinese as a foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–47.

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

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1226–1233.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pages 439–445.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. *arXiv preprint arXiv:1903.00138*.

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

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pages 117–128.

张宝林. 2008. “hsk 动态作文语料库” 的标注问题. 第五届中文电化教学国际研讨会, pages 363–370.

汪权彬and 谭营. 2020. 基于数据增广和复制的中文语法错误纠正方法. 智能系统学报, 15(1):99–106.

王辰成, 杨麟儿, 王莹莹, 杜永萍, and 杨尔弘. 2020. 基于transformer 增强架构的中文语法纠错方法. 中文信息学报, 34(6):106–114.

A 双重噪声化P_{rate}计算过程

针对同一句话，假设句子中包含n个字，为了使字粒度噪声化结果和双重噪声话结果保持一致，做出如下推导和假设：

【1】情况1：字粒度单次噪声化

记

$$X_i = \begin{cases} 1 & , \quad \text{第i个单词被修改} \\ 0 & , \quad \text{第i个单词未被修改} \end{cases} \tag{8}$$

其中X_i符合$X_i \sim b(1,p)$分布，这里p为每个单词被修改的概率，对于单次噪声化，文中该值设为0.3。那么共有$X = \sum_{i=1}^{n} X_i$ 个单词被随机加噪（修改）。因为每次加噪都是独立同分布的，所以$X \sim b(n,p)$，即$E(X) = np$

【2】情况2：字粒度和词粒度双重噪声化

分别记

$$Y_i = \begin{cases} 1 & , \quad \text{第一轮第i个单词被修改} \\ 0 & , \quad \text{第一轮第i个单词未被修改} \end{cases} \tag{9}$$

$$Z_i = \begin{cases} 1 & , \quad \text{第二轮第i个单词被修改} \\ 0 & , \quad \text{第二轮第i个单词未被修改} \end{cases} \tag{10}$$

为了方便计算，这里我们假设分词后句子中词数依然为n，同时第一次加噪和第二次加噪的概率P_{rate}相同，下面简写为q。与情况一类似，Y_i与Z_i分别符合$Y_i \sim b(1,q)$与$Z_i \sim b(1,q)$分布。那么理想情况下共有$K = \sum_{i=1}^{n}(Y_i + Z_i - Y_iZ_i)$ 个单词被随机加噪（修改）。因为每次加噪都是独立同分布的，所以$E(K) = 2nq - nE(Y,Z)$。对于$E(Y,Z)$，只有当Y_i与Z_i均为1时，$Y_iZ_i \neq 0$，且两次加噪也是独立同分布的，所以$E(Y_iZ_i) = E(Y_i)E(Z_i) = q^2$。故$E(K) = 2nq - nq^2$。

为了使情况【1】与情况【2】加噪单词数保持一致，于是有$np = 2nq - nq^2$，带入已知参数$p = 0.3$，解出$P_{rate} = q = 0.163$。