文章编号：1003-0077(2020)06-0106-09

基于 Transformer 增强架构的中文语法纠错方法

王辰成[1,2]，杨麟儿[2,3]，王莹莹[2,3]，杜永萍[1]，杨尔弘[2,3]

(1. 北京工业大学 信息学部，北京 100124；2. 北京语言大学 语言资源高精尖创新中心，北京 100083；
3. 北京语言大学 信息科学学院，北京 100083)

摘 要：语法纠错任务是自然语言处理领域的一项重要任务，近年来受到了学术界广泛关注。该任务旨在自动识别并纠正文本中所包含的语法、拼写以及语序错误等。该文将语法纠错任务看作是翻译任务，即将带有错误表达的文本翻译成正确的文本，采用基于多头注意力机制的 Transformer 模型作为纠错模型，并提出了一种动态残差结构，动态结合不同神经模块的输出来增强模型捕获语义信息的能力。受限于目前训练语料不足的情况，该文提出了一种数据增强方法，通过对单语语料的腐化从而生成更多的纠错数据，进一步提高模型的性能。实验结果表明，该文所提出的基于动态残差的模型增强以及腐化语料的数据增强方法对纠错性能有着较大的提升，在 NLPCC 2018 中文语法纠错共享评测数据上达到了最优性能。

关键词：语法纠错；多头注意力；动态残差结构；数据增强

中图分类号：TP391 **文献标识码**：A

Chinese Grammatical Error Correction Method Based on Transformer Enhanced Architecture

WANG Chencheng[1,2], YANG Liner[2,3], WANG Yingying[2,3], DU Yongping[1], YANG Erhong[2,3]

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;
2. Beijing Advanced Innovation Center for Language Resources, Beijing Language and Culture University, Beijing 100083, China; 3. School of Information Science, Beijing Language and Culture University, Beijing 100083, China)

Abstract: Grammatical error correction is an important task in the field of natural language processing, which has attracted wide attention in recent years. This paper regards grammatical error correction task as a translation task to translate the wrong texts into the right ones. We use the transformer model with multi-head attention mechanism as framework, and propose a dynamic residual structure to combine the outputs of different neural blocks dynamically to better capture semantic information. Due to the lack of training corpus, we propose a data augmentation method to generate the parallel data by corrupting a monolingual corpus. The experimental results show that the proposed method based on dynamic residuals and data augmentation has significantly improved the performance of error correction, achieving the best performance on NLPCC 2018 Chinese grammatical error correction task.

Keywords: grammatical error correction; multi-head attention; dynamic residual structure; data augmentation

0 引言

语法纠错(grammatical error correction，GEC)任务，旨在利用自然语言处理技术，自动识别并纠正非中文母语学习者书写的文本中所包含的语法错误、拼写错误、语序错误、标点错误等，是自然语言处理的一项重要任务。下面这对语句就是语法纠错任务的一个示例，每个输入对应一个输出，左侧输入的是一句可能带有语法错误的文本，右侧输出的是纠正语法错误后的结果，句中加粗的字是有修改的地方。

示例：这个软件让我们什么有趣的事都记录。→这个软件**能**让我们**把**有趣的事都记录**下来**。

目前语法纠错的方法有三种：第一种是基于规则的方法，针对具体的错误类型，制订特定的纠错规

收稿日期：2019-09-16 **定稿日期**：2019-11-20

基金项目：北京语言大学语言资源高精尖创新中心项目(TYZ19005)；国家语委信息化项目(ZDI135-105，YB135-89)

则对错误进行纠正，这种方法依赖于规则制定的好坏且只能修改特定的几种错误；第二种方法是基于统计的方法，通过字、词等相关信息的 Ngram 获取文本特征，对语言进行建模，选取合适的统计模型对文本进行纠错；第三种方法是基于深度学习的方法，利用词向量表示构建深度神经网络，在不考虑具体错误类型的情况，端到端地对文本中的错误进行纠正。其中比较常用的模型包括基于短语的统计机器翻译模型和基于 LSTM 或者 CNN 的神经机器翻译模型，但其共同的缺点是，认为一句话中的每个字或词都具有同等的重要性，无法有选择性进行关注。例如，当修改“能”字时，模型应该更关注“软件”以及“让我们”，而其他的字并没有过多的信息能够帮助其改正这个错误。

因此，我们采用基于多头注意力机制的 Transformer 序列生成模型[1]作为我们的纠错模型，并且提出了一种动态残差结构，能够增强模型挖掘文本语义信息的能力。由于中文语法纠错的训练语料过少，无法充分训练序列生成模型，我们提出了一种腐化语料的单语数据增强方法，能有效地扩大训练集的规模，并进一步提升模型的纠错效果。结合本文提出的两种方法，我们的模型在 NLPCC 2018 中文语法纠错共享任务上性能超过了所有其他模型，达到了最优的性能。

本文的组织结构如下：第 1 节介绍了语法纠错任务的相关工作；第 2 节详细描述了 Transformer 模型的细节以及提出的动态残差结构；第 3 节阐述了提出的通过腐化语料的单语数据增强方法；第 4 节介绍了本文的实验细节，并对实验结果进行分析，最后是结论。

1　相关工作

早期，解决语法纠错问题的主流方法是基于规则的方法[2-3]，但这些方法只能修正文本中特定的几种错误类型。为了解决更多类型的错误，Brockett 等人[4]首先提出了可以将语法纠错任务看作翻译任务的思想，并采用了一个基于统计机器翻译架构的语法纠错模型。

自 CoNLL-2013[5]和 CoNLL-2014[6]两个语法纠错共享任务的成功举办以来，该领域受到学者们的广泛关注，大量新颖且有效的方法被提出。谭咏梅等人[7]提出一种基于语料库的规则自动抽取方法并在此基础上提出了有限回退算法对英语文章进行语法错误检查及纠正。Junczys-Dowmunt 等人[8]率先使用一种特殊定制的统计机器翻译架构 Moses[9]，并结合了大规模的开放平行语料 Lang-8[10]。同年，Felice 等人[11]将基于规则的方法、统计机器翻译模型和语言模型三者相结合，进一步提高系统的纠错效果。

随着深度学习的广泛应用，神经机器翻译相关的方法也被引入到语法纠错领域中来。谭咏梅等人[12]针对冠词和介词错误提出了一种基于 LSTM 的序列标注 GEC 方法和一种 N-gram 的投票策略。Chollampatt 等人[13]使用多层卷积的序列生成模型，利用语言模型对最后生成的多个结果进行重新排序，成为第一个超过统计机器翻译方法的神经纠错模型。Grundkiewicz 等人[14]认为统计机器翻译方法仍具有无法替代的作用，故将统计机器翻译模型与基于循环神经网络的神经机器翻译模型相结合，并达到了很好的效果。

针对英语文本的语法纠错研究已经进行了长期的研究并取得优秀的成果，但中文的语法纠错才刚刚开始。王洁[15]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。龚小谨等人[16]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。Rao 等人[17]举办了 IJCNLP 2017 中文语法错误诊断公开评测任务，主要目的在于语法错误检测与识别，但没有涉及修正。CCF 国际自然语言处理与中文计算会议（NLPCC）在 2018 年的竞赛单元中，首次增加了中文语法错误修正任务[18]。这次竞赛的举办，给中文纠错的研究人员们提供了一个良好的平台。Fu 等人[19]将错误分成两类，简单错误（标点错误以及同音字错误）和复杂错误（其余所有错误），利用语言模型先将句子中的简单错误改正，再利用 Transformer 模型移除复杂错误。Zhou 等人[20]采用多模型平行结构，使用基于规则、基于统计和神经网络三大类模型，先在类别内进行低级组合得到类别候选，再对类别候选进行高级组合。Ren 等人[21]将切词后输入文本中的每个词语都利用 Subword 算法[22]拆分成子词单元，并采用了基于 CNN 的序列生成模型。

2　基于动态残差结构的 Transformer 模型

2.1　Transformer 模型

Transformer 模型是一种基于多头注意力机制

的序列生成模型，其核心结构如图1所示。编码器(Encoder)负责将输入文本编码为高维隐含语义向量，解码器(Decoder)依据上一步编码的输出，解码隐含语义向量为当前步骤的输出向量，每个步骤的输出向量对应一个字，所有步骤输出的字拼在一起得到最终输出的句子。

编码器 由 N 个完全一样的神经模块(Block)组成，模块之间输入和输出连接在一起。每一个模块都包含两个部分，多头注意力层以及前馈层。其中，多头注意力层 Multi-Head 是由多个注意力层拼接组成的，每个注意力层 Attention 采用的是范围点积 (scaled dot-product)，如式(1)、式(2)所示。

$$\mathrm{Att}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

$$\text{Multi-Head} = [\mathrm{Att}_1;\mathrm{Att}_2;\cdots;\mathrm{Att}_h] \tag{2}$$

其中，$\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 分别表示注意力层的查询矩阵(query)、键值矩阵(key)以及实值矩阵(value)，它们是由输入向量经过三个不同的线性层(linear)得到，即计算输入向量的自注意力。d_k 为模型的embedding层的第三维度大小，该因子是为了调节 $\boldsymbol{Q}$ 与 $\boldsymbol{K}$ 转置的内积大小，以防止过大的内积经过 Softmax 后向量分布不均匀。Multi-Head 为多个注意力的拼接。

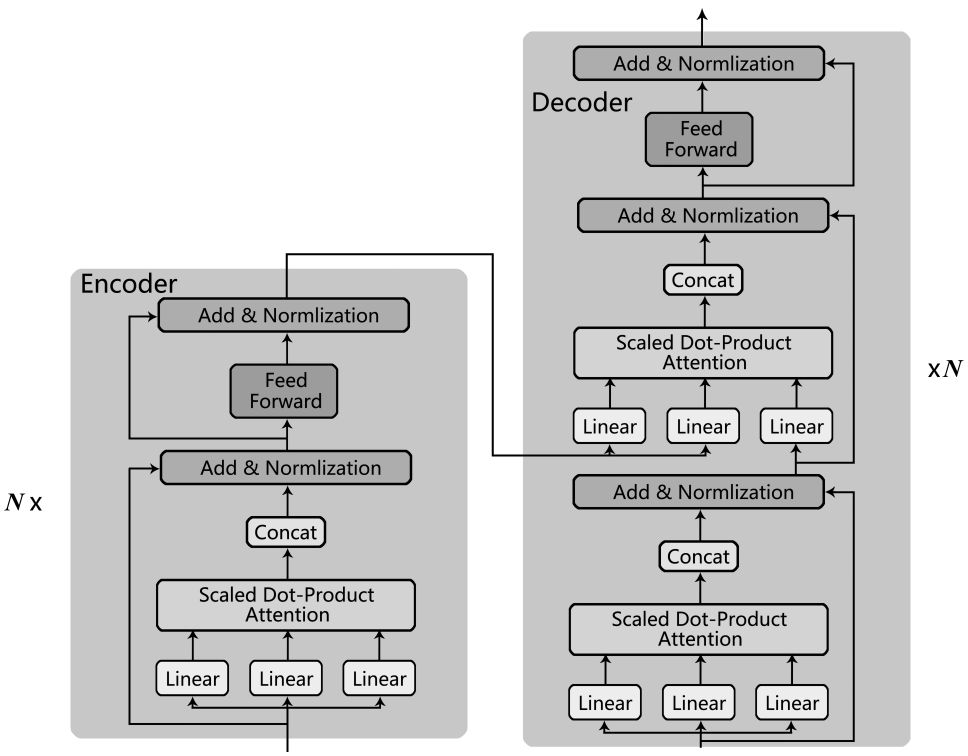

图1 Transformer 语法纠错模型

前馈层是由两个线性层串行连接而成的，其具有独立的权重与偏差，而且维度也不相同，能够进一步提取语义信息，如式(3)所示。

$$FF = \mathrm{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{3}$$

其中，W_1，b_1，W_2，b_2 分别为两个线性层的权重和偏差，其目的是将输入 x 的第三维先扩大到 W_1 第二维度的大小，再进行还原。

解码器 同样由 N 个完全一样的神经模块组成，除了与编码器中完全一样的两个部分以外，其还包含了一个额外的编解码多头注意力层。计算方式见式(1)，与多头注意力层不同点在于 $\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 的取值。其中，$\boldsymbol{Q}$ 与 $\boldsymbol{K}$ 都是编码器的输出向量，而 $\boldsymbol{V}$ 是解码器的多头注意力层的输出，即计算的是编码器与解码器向量间的注意力。

编码器与解码器中，每个独立的层后都有一个归一化层以及一个残差结构。归一化层能够将经过的向量值映射到0-1之间，加快模型的收敛速度；残差结构的作用是使得模型深度过深时，梯度不会为0。

2.2 动态残差结构

模型中的神经模块(block)可以简化地看作为图2的结构。所以，每个模块的输出如式(4)所示。

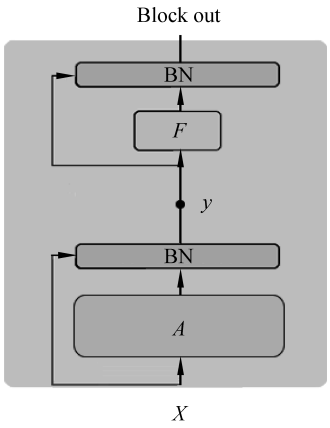

图 2　简化的神经模块

$$y = \mathrm{BN}(A(x)+x), \quad \mathrm{out} = \mathrm{BN}(F(y)+y) \tag{4}$$

其中，BN 表示归一化函数，A 表示注意力操作，F 对应的是前馈层的线性变换，对式(4)求导得到式(5)。

$$\begin{aligned}\frac{\partial \mathrm{out}}{\partial y} &= \frac{\partial \mathrm{BN}}{\partial y} * \left(\frac{\partial F}{\partial y}+1\right),\\ \frac{\partial y}{\partial x} &= \frac{\partial \mathrm{BN}}{\partial x} * \left(\frac{\partial A}{\partial x}+1\right),\\ \frac{\partial \mathrm{out}}{\partial x} &= \frac{\partial \mathrm{out}}{\partial y} * \frac{\partial y}{\partial x}\end{aligned} \tag{5}$$

可以看到，虽然模块中有两个残差结构，但求导后依旧存在乘法因子$\frac{\partial \mathrm{BN}}{\partial y} * \frac{\partial \mathrm{BN}}{\partial x}$，随着模块的增多，这个因子会被连乘累积起来，导致模型仍旧存在梯度消失的可能。所以，我们将所有神经模块的输出的累加作为最终的输出，如图 3 所示，这样损失函数的梯度不会因为模型深度增加而消失。

在训练的过程中，每个神经模块所学习到权重也不尽相同。但是，直接采用最后一个神经模块的输出，作为编码器或者解码器整体的输出可能会损失部分语义的信息。

受到 ELMo[23] 工作启发，我们认为在不同的神经模块中学习到的知识可能有互补的作用，即高层的神经模块的状态可以捕捉到词语意义中和语境相关的特征，而低层的可以找到语法方面的特征，将所有模块的输出动态地结合到一起，可以表达更加丰富的语义信息。动态结合的方式如图 3 所示，具体计算如式(6)所示。

$$\mathrm{out} = \sum_{i=1}^{N} \mathrm{Block}_i \cdot \boldsymbol{W}_i \tag{6}$$

其中，$\boldsymbol{W}_i \in R^{1\times 1}$，即计算每个神经模块的输出与权值的点乘，再将所有的乘积求和，式(6)中的变量 $\boldsymbol{W}_i$ 通过反向传播算法学习得到。这种动态的残差结构，可以应用到 Transformer 模型的编码器或者解码器端，不仅能够帮助模型捕获更加丰富的语义信息，其中的残差结构还可以减少因为模型过深而带来的梯度消失的问题，帮助深度神经网络更好地训练。

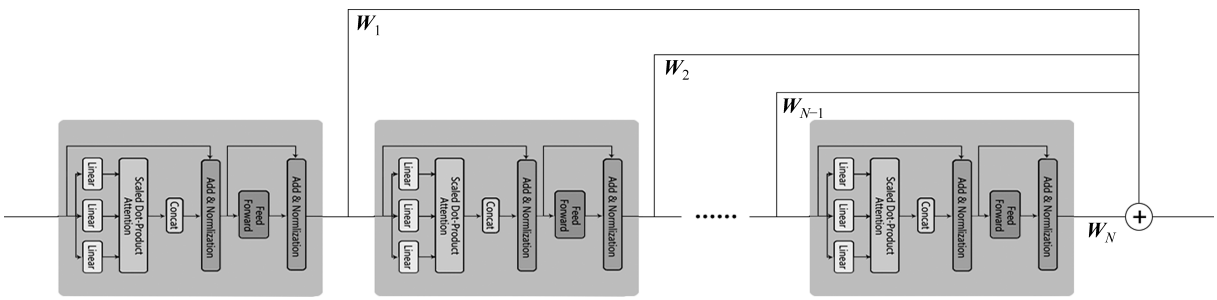

图 3　动态残差结构

3　基于腐化语料的单语数据增强方法

对于深度神经网络的学习，最有效的方法就是扩大数据规模，但受限于目前中文语法纠错领域训练语料不足的情况，没有更多的并行数据(一句错误的语句和一句纠正的语句)可供模型训练。但是，互联网中存在着大量的中文单语数据，即完全正确的中文语句。在这些容易获取且完全正确的单语语料中，合理地添加错误，即可得到大量的语法纠错并行语料。这些生成的语料中，可以包括任何多样化的错误种类以及语法现象。

示例 1

错误：带学生去哈尔滨，这件事只有由你负责吗？

正确：带学生去哈尔滨这件事是只由你负责吗？

示例 2

错误：她也就是说爱撒娇。

正确：也就是说她爱撒娇。

上面两对纠错数据中，第一对示例应将“，”和“有”删除，并添加“是”字，第二对示例需要将“她也就是说”替换为“也就是说她”。因此，我们认为可以简单地将人们常犯的错误按照添加删除替换的规则区分为多字错误缺字错误以及替换错误。

针对上述三种错误类型，我们设计了一种腐化算法，可以根据所需的错误类型比例，对单语语料进行造错，具体实现如表1所示。

表1 基于腐化语料的单语数据增强算法

输入：单语语料 C_m，腐化词语的比例 E_{rate}，多字错误比例 P_u，缺字错误比例 P_m，替换错误比例 P_r
输出：腐化后的语料 C

```
Begin
  1. 加载单语语料，利用 jieba 分词工具对单语语料进行分词，去除超过 80 个词语的的文本。
  2. 统计所有分词后的数据所包含的不同的词语，将全部词语的集合记为词表 V.
  3. for all sentence S in C_m do                    /* 遍历单语语料 C_m 中的每个语句 S */
       for all words w in S do                       /* 遍历语句 S 中的每个词语 w */
            获取 1 到 100 的一个随机数 rand
         if rand>E_rate then                         /* 控制语料中腐化单词的比例 */
            continue                                 /* 大于 E_rate 就不进行腐化操作了 */
         end if
         再次获取 1 到 100 的一个随机数 rand
         if rand<P_u then                            /* P_u%的概率添加一个词语，造成多字错误 */
            在 w 左边添加一个词表 V 中的随机词语 w'
         else if rand<P_u+P_m then                   /* P_m%的概率删除一个词语，造成缺字错误 */
               删除当前词语 w
            else if rand<P_u+P_m+P_r then            /* P_r%的概率替换一个词语，造成替换错误 */
            替换当前词语 w 为词表 V 中的随机词语 w'
         end if
       end for
       将腐化后的语句添加到语料 C 中
     end for
  4. 输出腐化后的语料 C
End
```

我们统计了训练数据中错误语句到正确语句的最短编辑距离（最少的添加、删除、替换的操作的数量），并除以正确语句包含的词语总数，得到整份语料的错误率，约为29%。我们利用表1的算法对全部的单语数据进行腐化，其中，E_{rate}设置为30%，与训练语料的错误率保持相似，设置多字、缺字以及替换错误的比例为1∶1∶1. 我们将腐化后的语句与分词后的语句组成新的语法纠错平行语料。

数据增强的结果如表2所示，可以看到jieba分词工具很好地对原始的语句进行了分词，而腐化的方法也将原本完全正确的语句，修改成带有错误的语句。虽然腐化的结果有些不符合人类的语言习惯，但我们会在4.4节验证腐化语料的单语数据增强方法的有效性。

表2 数据增强结果示例

原始语句	列车为前往机场的乘客提供了行李架
分词后的语句	列车/为/前往/机场/的/乘客/提供/了/行李架/
腐化后的语句	列车/本应/前往/机场/的/朝兴/乘客/猫爪草/了/行李架

4 实验与分析

4.1 实验数据及预处理

本文所采用的平行语料包括NLPCC官方提供的Lang-8汉语数据集[①]以及HSK数据集。其中

① http://tcci.ccf.org.cn/conference/2018/dldoc/trainingdata02.tar.gz

Lang-8 语料采集自 lang-8.com 作文批改平台[10]，来自不同国家的学习者在该平台上进行写作，相对应的精通该语言的其他用户会对这批数据进行修改，每一篇文章都会有不等数量的用户进行修改，所以，相同的错误可能会有多种改正的结果。HSK 数据集是北京语言大学构建的动态作文语料库[24]中的数据，采集自 1992—2005 年 HSK 作文考试的答卷，涉及约 50 种错误类型。测试集选用的是 NLPCC2018 年公开评测比赛的测试集①。文中数据增强方法所使用的中文单语语料采集自中文维基百科的数据②。所有数据集的规模如表 3 所示。

表 3　数据集规模统计

语料名称	句子	句子(修改)	词语	词语(修改)
Lang-8	1 220 734	1 097 233	15 650 807	14 587 611
HSK	156 870	96 049	2 713 134	1 913 009
中文维基百科	7 423 942	—	146 033 068	—
NLPCC 2018 测试集	2 000	—	39 507	—

全部平行语料共有 1 349 769 对训练语句，我们去除其中没有修改的句对，即错误语句与修改语句完全一样的训练数据，剩余语料包括 1 178 397 对训练语句。随后从剩余语料中随机选择 5 000 条数据作为验证集。

分词　中文语料首先需要进行分词，我们选择 jieba 分词工具对全部语料进行分词，测试集已经被官方用 pkunlp 工具包③分词，我们去除掉测试集中词语之间的空格，将语料还原到未分词状态，再使用 jieba 进行分词，这样做的目的是统一训练集和测试集的分词方法，能够使得测试过程得到最好的结果。模型对测试集进行修改完成后，再用 pkunlp 进行分词。

子词单元　子词算法是一种字节对编码算法(byte pair encoding，BPE)，其能够将罕见的词语分割成多个频繁出现的子词单元。这种算法应用到语法纠错任务中，有助于解决稀有词和未登录词(out of vocabulary，OOV)问题。

利用子词算法拆分后的结果如表 4 所示，“@@”符号表示当前子词单元与后一个子词单元同属一个词语的关系。我们在训练模型之前，利用子词算法对全部数据进行处理，而在测试阶段，将模型输出中的子词单元还原为拆分之前的词语。

表 4　子词单元拆分示例

原始语句	自己学识浅薄孤陋寡闻无法如愿以偿，只有拍案兴叹！
分词后的语句	自己/学识/浅薄/孤陋寡闻/无法/如愿以偿/，/只有/拍案兴叹/！
子词算法拆分后的语句	自己/学识/浅@@/薄/孤@@/陋@@/寡@@/闻/无法/如愿以偿/，/只有/拍@@/案@@/兴@@/叹/！

4.2　模型参数设置

本文 Transformer 模型的实现使用的是Fairseq 开源代码库④。模型具体的超参数设置为：源端词嵌入矩阵和目标端词嵌入矩阵采用同样的 512 维大小的词表，且目标端的输入与输出词嵌入矩阵共享权重。编码器与解码器各包含 6 个神经模块，每个模块的多头注意力层有 8 个注意力头，前馈层 $\boldsymbol{W}_1$ 的第二维度大小为 2 048。我们使用 Adam 优化器训练 Transformer 模型，学习率初始值为 1×10^{-7}，在前 4 000个 batch 的训练中，线性的增长到 5×10^{-4}，后续的训练中逐步下降，直到训练结束，dropout 设置为 0.3。

在训练的过程中，每训练完一次就在验证集上进行验证，取最优的 5 个结果的模型权重，计算权重的平均值作为最终的模型的权重设置。在解码阶段，我们设置 beam search 大小为 12，并取最好的结果作为模型最终的输出。

4.3　评价指标

语法纠错任务的目标是改正句子中的错误单词，评价依据模型对错误语句的编辑与标准编辑集合的匹配程度。评价指标为，准确率(Precision，P)、召回率(Recall，R) 以及 $F_{0.5}$。假设，e 是模型对错误文本修改的编辑集合，ge 是该错误文本的标准编辑集合，具体的计算如式(7)～式(9)所示。

$$P=\frac{\sum_{i=1}^{N}|e_i\cap \mathrm{ge}_i|}{\sum_{i=1}^{N}|e_i|} \tag{7}$$

① http://tcci.ccf.org.cn/conference/2018/dldoc/tasktestdata02.zip

② https://storage.googleapis.com/nlp_chinese_corpus/wiki_zh_2019.zip

③ http://59.108.48.12/lcwm/pkunlp/downloads/libgrassui.tar.gz

④ https://github.com/pytorch/fairseq

$$R=\frac{\sum_{i=1}^{N}|e_i\cap \mathrm{ge}_i|}{\sum_{i=1}^{N}|g_i|} \tag{8}$$

$$F_{0.5}=\frac{(1+0.5^2)\times R\times P}{0.5^2\times P+R} \tag{9}$$

其中，$|e_i\cap \mathrm{ge}_i|$表示模型针对句子 i 修改的编辑集合与标准编辑集合的匹配数量，具体计算方法如式(10)所示。

$$|e_i\cap \mathrm{ge}_i|=\{e\in e_i\mid \exists g\in g_i,\mathrm{match}(e,g)\} \tag{10}$$

语法纠错任务选择 $F_{0.5}$ 作为评价指标而非 F_1 值的原因在于，对于模型纠正的错误中，更加看重编辑的准确性而非更多的编辑数量，所以将准确率的权重定为召回率的两倍大小，以期得到一个更加优质的纠错模型。我们选择公开的 MaxMatch(M2)工具包计算 $F_{0.5}$，其可高效地搜索一组模型的编辑，以便最大限度地匹配标准编辑集合。

4.4 实验结果

我们分别做了5组实验，来验证本文所提出的方法的有效性。第一组实验仅仅使用 Transformer 模型进行纠错；第二组实验是在 Transformer 的编码器端(Encoder)加上本文所提出的动态残差结构；第三组实验在 Transformer 的解码器端(Decoder)加上动态残差结构；第四组是在编码器端和解码器端都加入动态残差结构(ALL)；第五组是在第三组实验的基础上，使用了本文提出的数据增强方法，实验结果如表5所示。

表5 不同改进对模型性能的影响

Model	P	R	$F_{0.5}$	$\Delta F_{0.5}$
Transformer	37.83	20.66	32.44	—
+动态残差结构(Encoder)	36.81	19.21	31.11	−1.33
+动态残差结构(Decoder)	39.10	21.49	33.60	+1.16
+动态残差结构(ALL)	38.53	20.69	32.86	+0.42
+动态残差结构(Decoder)&数据增强	39.43	22.80	**34.41**	**+1.97**

由表5可以看到，基于多头注意力机制的 Transformer 已经达到很高的 $F_{0.5}$ 值。动态残差组的第三第四组实验相对于第一组结果都有明显的提升，但第二组的 $F_{0.5}$ 值略有降低，对比这三组的结果，我们发现在解码器端添加动态残差结构可以很好地提升模型的表现，但是在编码器端如动态残差却会损害模型的纠错能力，这也就说明了第四组实验中，$F_{0.5}$ 值的提升主要贡献都来自于解码器端而非编码器端。第五组实验是在第三组实验上又加入了数据增强的方法，使得模型的性能达到了最优。

同时，我们与 NLPCC2018 中文语法纠错共享任务的前三名团队进行了结果的比较，如表6所示，其中，“4 ens.”表示4个模型集成的结果，“LM”表示利用了额外的语言模型。从表6可以看到，我们模型的 $F_{0.5}$ 值优于所有其他工作，而且该结果仅仅是我们使用单一模型得到，没有任何模型的集成和语言模型的使用。

表6 与 NLPCC 2018 评测 TOP3 工作的性能的对比

	P	R	$F_{0.5}$
AliGM(4 ens.)[20]	41.00	13.75	29.36
YouDao(5 ens. + LM)[19]	35.24	18.64	29.91
CS2S(4 ens.)[21]	47.63	12.56	30.57
Our Model(Single)	39.43	22.80	**34.41**

我们同时也分析了数据增强方法的影响，包括对测试集中六种具体的错误类型的 $F_{0.5}$ 值的影响。从图4(a)中可以看出，使用数据增强方法之后，在名词、符号、语序和其他错误四种错误类型上有着显著提高，而在介词错误与拼写错误上却没有有效提升。图4(b)显示，数据增强方法产生了更多的名词、语序与其他类型的错误，从而增多了这三种错误的比例，降低了介词与拼写错误的比例，比例上增减的变化与对应的 $F_{0.5}$ 的变化相吻合。因此，数据增强策略应该是能够同比增加不同错误类型的数量。

5 总结

本文的工作将语法纠错任务看作是翻译任务，利用基于多头注意力机制的 Transformer 对错误文本进行改正，同时提出了一种动态残差结构，可以分别结合到 Transformer 的编码器与解码器端，用以增强模型捕获丰富语义信息的能力。受限于训练数据过少的情况，我们还提出了一种腐化语料的单语

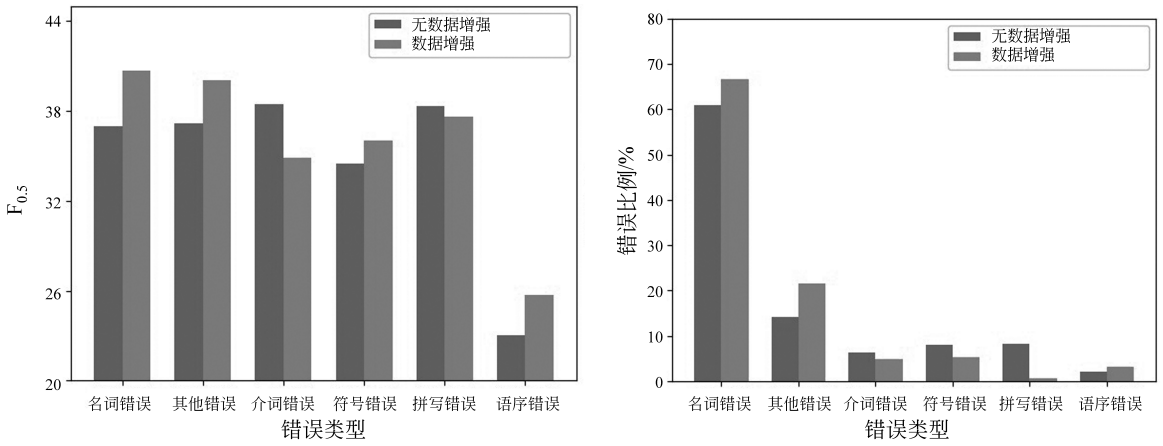

图 4　使用数据增强方法对不同错误类型的影响

数据增强方法，扩充了训练集的规模。这种数据增强的方法可以在任何领域或者语言的单语语料上使用。通过实验进一步验证了本文提出的模型增强与数据增强方法的有效性，在 NLPCC 2018 中文语法纠错共享评测任务上达到了最优的性能。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of Advances in Neural Information Processing Systems, 2017: 5998-6008.

[2] Bustamante F R, León F S. GramCheck: A grammar and style checker[C]//Proceedings of the 16th Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics, 1996: 175-181.

[3] Heidorn G E, Jensen K, Miller L A, et al. The EPISTLE text-critiquing system[J]. IBM Systems Journal, 1982, 21(3): 305-326.

[4] Brockett C, Dolan W B, Gamon M. Correcting ESL errors using phrasal SMT techniques[C]//Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2006: 249-256.

[5] Ng H T, Wu S M, Wu Y, et al. The CoNLL-2013 shared task on grammatical error correction[C]//Proceedings of the 17th Conference on Computational Natural Language Learning: Shared Task, 2013: 1-12.

[6] Ng H T, Wu S M, Briscoe T, et al. The CoNLL-2014 shared task on grammatical error correction[C]//Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task, 2014: 1-14.

[7] 谭咏梅，王晓辉，杨一枭. 基于语料库的英语文章语法错误检查及纠正方法[J]. 北京邮电大学学报，2016，39(4):92-97.

[8] Junczys-Dowmunt M, Grundkiewicz R. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation[C]//Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task, 2014: 25-33.

[9] Koehn P, Hoang H, BirchA, et al. Moses: Open source toolkit for statistical machine translation[C]//Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007: 177-180.

[10] Mizumoto T, Komachi M, Nagata M, et al. Mining revision log of language learning SNS for automated Japanese error correction of second language learners[C]//Proceedings of 5th International Joint Conference on Natural Language Processing, 2011: 147-155.

[11] Felice M, Yuan Z, Andersen Ø E, et al. Grammatical error correction using hybrid systems and type filtering[C]//Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task, 2014: 15-24.

[12] 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.

[13] Chollampatt S, Ng H T. A multilayer convolutional encoder-decoder neural network for grammatical error correction[C]//Proceedings of 32nd AAAI Conference on Artificial Intelligence, 2018.

[14] Grundkiewicz R, Junczys-Dowmunt M. Near human-level performance in grammatical error correction

with hybrid machine translation[J]. arXiv: 1804.05945, 2018.

[15] 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.

[16] 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.

[17] Gaoqi R, Zhang B, Endong X, et al. IJCNLP-2017 task 1: Chinese grammatical error diagnosis[C]//Proceedings of the IJCNLP 2017, Shared Tasks. 2017: 1-8.

[18] Zhao Y, Jiang N, Sun W, et al. Overview of the NLPCC 2018 shared task: Grammatical error correction[C]//Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2018: 439-445.

[19] Fu K, Huang J, Duan Y. Youdao's winning solution to the NLPCC-2018 Task 2 challenge: A neural machine translation approach to Chinese grammatical error correction[C]//Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2018: 341-350.

[20] Zhou J, Li C, Liu H, et al. Chinese grammatical error correction using statistical and neural models [C]//Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2018: 117-128.

[21] Ren H, Yang L, Xun E. A Sequence to sequence learning for Chinese grammatical error correction [C]//Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2018: 401-410.

[22] Sennrich R, Haddow B, Birch A. Neural machine translation of rare words with subword units[J]. arXiv preprint arXiv:1508.07909, 2015.

[23] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 2227-2237.

[24] 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.

王辰成(1995—),硕士研究生,主要研究领域为自然语言处理。
E-mail: hsamswang@gmail.com

王莹莹(1993—),博士研究生,主要研究领域为自然语言处理和智能计算机辅助语言学习。
E-mail: ying_y_wang@126.com

杨麟儿(1983—),通信作者,博士,主要研究领域为自然语言处理和智能计算机辅助语言学习。
E-mail: yangtianlin@blcu.edu.cn