

# 基于Transformer局部信息及语法增强架构的中文拼写纠错方法

段建勇 袁阳 王昊<sup>†</sup>

北方工业大学信息学院, 北京 100043; <sup>†</sup> 通信作者, E-mail: wanghaomails@gmail.com

**摘要** 针对中文拼写纠错, 提出两种新的改进方法。其一, 在Transformer注意力机制的基础上, 添加高斯分布的偏置矩阵, 用于提高模型对局部文本的关注程度, 加强对错误文本中错误字词和周边文字的信息提取。其二, 使用ON\_LSTM模型, 对错误文本表现出的特殊语法结构特征进行语法信息提取。实验结果表明, 所提出的两种方法均能有效提高准确率和召回率, 并且, 将两种方法融合后的模型取得最高F1值。

**关键词** 拼写纠错; Transformer模型; 局部信息; 语法增强

## Chinese Spelling Correction Method Based on Transformer Local Information and Syntax Enhancement Architecture

DUAN Jianyong, YUAN Yang, WANG Hao<sup>†</sup>

School of Information Science and Technology, North China University of Technology, Beijing 100043;  
<sup>†</sup> Corresponding author, E-mail: wanghaomails@gmail.com

**Abstract** Two new methods for improving Chinese spelling correction are proposed. The first one is to add Gaussian Bias matrices to the Transformer's attention mechanism, which is used to improve the model's attention to local text and to extract information from the wrong words and the surrounding text in the error text. Secondly, the ON\_LSTM model is used to extract grammatical information on the special grammatical structure features exhibited by the error text. The experimental results show that both methods are effective in improving accuracy and recall, and the model after fusing the two methods achieves the highest F1 value.

**Key words** spelling correction; Transformer model; local information; grammatical enhancement

在文字编写、内容审核以及文本识别等多个场景中, 经常出现两种文本错误。比如, 由OCR的错误识别或“五笔”输入法的错误输入导致的同形字错误; 或者, 在拼音输入法下, 由错误拼写导致的同音字错误。拼写纠错任务用于检测并纠正文本中出现的错误, 可以在很大程度上解决上述两类文本错误问题。然而, 目前关于中文纠错的研究较少, 缺少有效的通用方法。

拼写纠错是自然语言处理(natural language processing, NLP)领域的一个重要任务。在20世纪60年代, 已经有学者对英文纠错展开研究, 其中比

较著名的是1960年IBM在IBM/360和IBM/370上, 用UNIX实现的一个TYPO英文拼写检查器<sup>[1]</sup>。此后, 文本纠错任务研究取得长足的进展, 涌现很多优秀的算法模型。

纠错任务需要有监督的语料进行模型训练, 这种局限性往往导致在训练过程中没有充足的文本纠错语料可用。N-gram模型很好地解决了这一弊端, 可利用大量无监督的语料来训练统计语言模型<sup>[2]</sup>。N-gram是最简单、最常用的模型, 但该模型在使用困惑集对文本内部的字符进行替换时, 没有考虑上下文的语义信息, 经常造成虽然临近词正确, 但放

在整个句子中却不符合逻辑的情况,导致结果得分不高。

随着机器学习的逐渐兴起,越来越多的学者把深度学习的方法运用到纠错任务中。一般将纠错任务当作文本生成任务,此场景下应用最广泛的结构就是序列到序列结构(Seq2Seq)。比较常用的是基于LSTM的Seq2Seq模型<sup>[3]</sup>和Transformer模型<sup>[4]</sup>,这两种模型巧妙地将错误句子作为输入语句,将正确的语句作为输出语句进行训练,得到错误文本与正确文本之间的对应关系,从而实现文本纠错。深度学习的优点在于,模型可以学习到文本的深层信息,加强对文本语义的理解,也加深对文本结构的学习,对获取推荐修改选项有更好的帮助。

可以将拼写纠错任务视为由错误句子产生正确句子的生成任务。在目前常用的生成模型中,RNN模型<sup>[3]</sup>在数多的生成任务中表现出色,在文本生成和机器翻译等领域获得广泛的应用。但是,经典的RNN模型在处理长距离文本时,容易出现梯度消失和梯度爆炸的问题,所以本文采用Transformer模型<sup>[4]</sup>,对中文文本中的同音和同形字错误进行纠错处理。

在文本纠错过程中,应注意以下两个方面。1)与英文不同,中文文本字符之间没有空格,且部分字符需要与另外的字符结合才具有一定的意义,所以在处理中文文本时,一般需要先对中文进行分词,将文本拆分成字或词,从而形成一个独立的单位。2)中文中没有“字错误”这种说法,也就是说,错误只会出现在词或句子级别产生。因此,错误字符以及错误字符周边的信息是需要重点关注的内容,Yang等<sup>[5]</sup>提出一种关注局部信息的方法,将局部性建模设计为一种可学习的高斯偏差。本文采用上述方法,在局部范围内加强对错误文本特征的提取以及对错误规律的发现能力。

拼写纠错模型以生成模型为最常见<sup>[3-4,6]</sup>,训练数据为原语句与目标语句构成的语对,原语句编码后,解码生成目标语句。在纠错任务中,原语句与目标语句之间可能只有一个或几个字符有所不同,在整个语句中的字符数量占比很小,模型训练往往达不到很好的效果。Wang等<sup>[7]</sup>通过构造大量的训练数据,使该问题得到缓解。另外,由于错误字符占比小,使得原语句与目标语句之间的文本编辑距离很小,但仍有可能导致原语句与目标语句相比,在语法结构上发生较大的变化。针对此问题,本文

在原始模型的基础上,采用Shen等<sup>[8]</sup>提出的ON\_LSTM方法,对LSTM结构进行修改,使其能够考虑到语句的语法结构对纠错效果的影响,同时将ON\_LSTM模型作为Transformer模型的输入层,帮助模型学习语句的语法结构。

## 1 模型结构

### 1.1 高斯分布约束的自注意力机制

Transformer的核心是使用自注意力机制,该机制因高效的并行性及长距离信息依赖性,在机器翻译、文本生成和对话系统等领域有重要的作用,获得广泛的关注。该模型摆脱了传统RNN模型长距离信息利用不充分、梯度消失和梯度爆炸的问题,已应用于各种NLP任务中。

给定句子 $x=\{x_1, x_2, \dots, x_n\}$ ,经过词向量层及位置编码层后,由3个权重矩阵分别生成query  $Q \in \mathbb{R}^{(l \times d)}$ ,key  $K \in \mathbb{R}^{(l \times d)}$ 和value  $V \in \mathbb{R}^{(l \times d)}$ 。每一层的注意力结果可以通过下式计算:

$$\text{Content}=\text{ATT}(Q, K)V, \quad (1)$$

其中,ATT(·)是点乘计算公式:

$$\text{ATT}(Q, K)=\text{softmax}(\text{scores}), \quad (2)$$

$$\text{scores} = \frac{QK^T}{\sqrt{d}}, \quad (3)$$

$\sqrt{d}$ 是固定因子, $d$ 是模型的隐藏层维度,scores是点乘计算得出的得分。

实际上,纠错任务的特殊性在于错误文本与正确文本之间的差别很小,而模型关注的范围过于宽泛,所以模型的重点关注范围是错误文本周边一定范围内的字符,提取错误字符特征,加强模型对纠错任务的适应能力。

本文采用Yang等<sup>[5]</sup>局部模型的思想,在自注意力得分的基础上,添加一个类似高斯分布的二次函数偏置项,用于减少对非错误部分的关注。如图1所示,在原始分布的基础上添加以错误字符“柏”为中心的高斯分布,使模型能够加强对该字符以及周边字符的注意力,并且该高斯分布的参数是可训练的。共有两种分布模式:高斯分布和强化的高斯分布。这两种分布模式都重点关注错误部分的信息,区别在于强化的高斯分布减少了对错误点的预测,增加对周边信息的关注,以便提高模型的召回率。

### 1.2 局部关注模型(local focus, LF)

计算注意力得分后,将偏置 $G$ 加入得分中,用

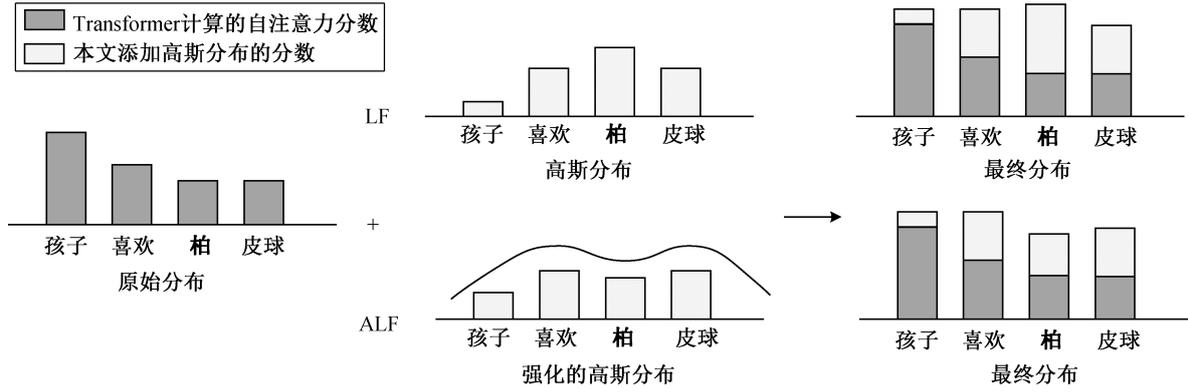


图 1 两种不同的高斯分布

Fig. 1 Two different Gaussian distributions

于掩盖自注意力的部分得分:

$$\text{ATT}(\mathbf{Q}, \mathbf{K}) = \text{softmax}(\text{scores} + \mathbf{G}), \quad (4)$$

其中,  $\text{scores}$  由式(3)计算得到,  $\mathbf{G}$  是添加的偏置项, 计算方法如下:

$$G_{i,j} = -\frac{(j-P_i)^2}{2\sigma_i^2}, \quad (5)$$

其中,  $\sigma_i$  表示标准差,  $\sigma_i = \frac{D_i}{2}$ ;  $D_i$  是一个窗口的大小;  $P_i$  是预测错误的位置;  $G_{i,j} \in [0, -\infty]$ 。  $P_i$  和  $D_i$  的计算方法如下:

$$\begin{bmatrix} P_i \\ D_i \end{bmatrix} = I \cdot \text{sigmoid} \left( \begin{bmatrix} p_i \\ z_i \end{bmatrix} \right). \quad (6)$$

参数  $I$  将标量  $p_i$  和  $z_i$  投影到 0 到句子长度之间,  $P_i$  依赖  $p_i$  和  $z_i$  分别进行计算。值得注意的是, Anastasopoulos 等<sup>[9]</sup>提出 Transformer 在不同的文本层中编码信息是有所区别的, 在较低的层, 模型对主要文本的语法结构编码, 而在高层, 模型对语义信息编码。为了加强模型对文本底层的结构信息编码, 仅在编码端和解码端的第一层使用该方法, 在其他层中, 编码方式与常规的自注意力保持一致。考虑到如果在所有的编码层中全部进行计算, 运行时间会变长, 故模型只在编码端和解码端的最底层添加偏置, 以期在效率上达到平衡。

### 1.2.1 中心位置预测

通过一层前线传播神经网络, 将 query 矩阵  $\mathbf{Q}_i$  转化为位置隐藏状态矩阵, 然后通过线性投影  $\mathbf{U}_p \in \mathbb{R}^d$ , 将隐藏状态矩阵映射到标量  $p_i$ :

$$p_i = \mathbf{U}_p^T \tanh(\mathbf{W}_p \mathbf{Q}_i), \quad (7)$$

其中,  $\mathbf{W}_p \in \mathbb{R}^{d \times d}$  是可训练的模型参数。

### 1.2.2 查询窗口

查询窗口  $z_i$  的计算方法如下:

$$z_i = \mathbf{U}_d^T \tanh(\mathbf{W}_p \mathbf{Q}_i), \quad (8)$$

其中,  $\mathbf{U}_d \in \mathbb{R}^d$  是可训练的线性投影参数矩阵, 式(7)和(8)共享同一参数  $\mathbf{W}_p$ , 这样可以简化运算, 同时用不同的  $\mathbf{U}_d$  和  $\mathbf{U}_p$  来计算窗口范围及中心点位置。

### 1.3 强化局部关注模型(advanced local focus, ALF)

在计算注意力得分时, 通常需要对周边信息更多的关注, 并减少对本身信息的关注。所以, 本文对 LF 模型进行改进, 在式(5)的基础上添加偏置  $b$  以及取绝对值的操作, 并设置  $b$  为 0.1,  $a$  为 0.1。计算方法如下:

$$G_{i,j} = -\frac{|a(j-P_i)^2 - b|}{2\sigma_i^2}. \quad (9)$$

与原始模型相比, 该模型减少了对本身及远距离内容的关注, 并加强周边范围内容的计算, 让模型学习更多错误的规律, 提升纠错效果。

### 1.4 语法强化

错误文字会在很大程度上导致语法结构被破坏。如图 2 所示, 在例句“孩子喜欢拍皮球”中, 如果输入时将“拍”错误输入成“柏”, 可以明显地看出错误句子的语法结构发生较大的变化。为了提取被破坏的语法结构信息, 本文采用 Chollampatt 等<sup>[6]</sup>中层级结构的 LSTM, 并称之为 ON\_LSTM。

### 1.5 模型结构

模型的总体结构如图 3 所示, 在 Transformer 结构外添加 ON\_LSTM 层, 将其输出结果与输入信息相加后送入 Transformer 模型。在左侧, 我们额外添加一层编码层。在内部的自注意力部分, 添加高斯

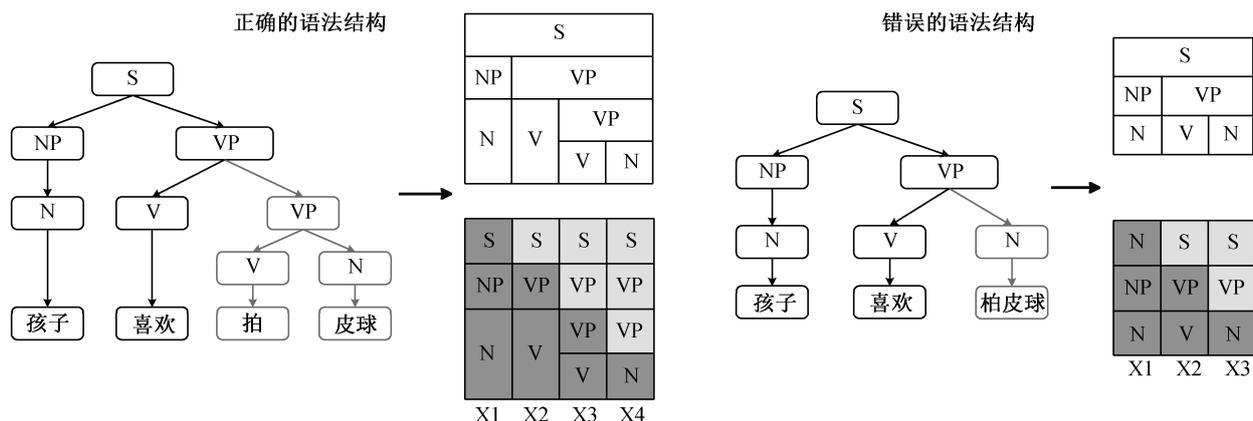


图2 因错误字引发的语法结构变化实例  
Fig. 2 An instance of a syntax change caused by an error word

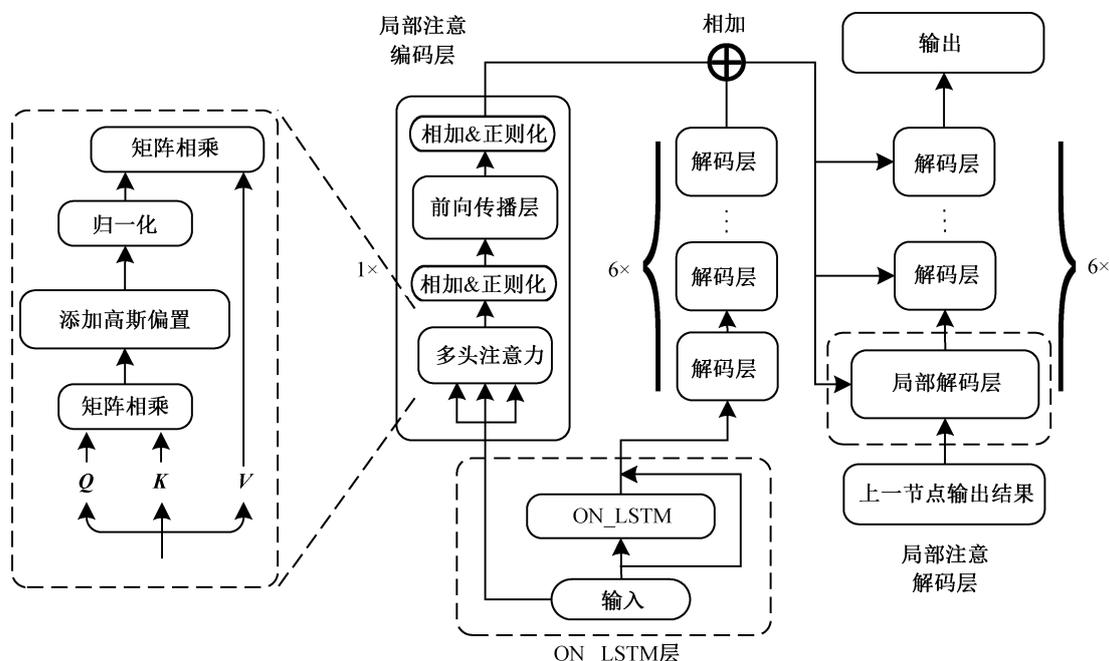


图3 模型的总体结构  
Fig. 3 Overall structure of the model

偏置, 增加对错误部分信息的获取。最后, 将添加的编码层及右侧常规的6个Transformer编码层的输出信息进行融合, 送入解码端进行解码。不同于传统的LSTM结构, ON\_LSTM结构在编码过程中通过控制高层信息和低层信息的更新频率来学习句子的语法结构。本文将ON\_LSTM产生的结构信息与原文的文本信息相加, 最后将得到的信息送入Transformer的编码层, 进行联合编码。这种结构类似Transformer中编码部分的残差模块, 意在加强数据的流通性, 减少在深层模型中梯度消失的问题。

## 2 实验

### 2.1 数据

训练数据分为两部分: 一部分是Wang等<sup>[7]</sup>提供的约27万条纠错数据集, 其中包括同音字和同形字错误; 另一部分来自文献[10]。测试集同样来自公开评测任务中提供的评测数据集, 具体信息如表1所示。由于数据全部是繁体中文, 为了适应简体中文任务, 我们使用开源工具OpenCC (<https://github.com/BYVoid/OpenCC>)将繁体中文转化为简体中文。

表 1 实验数据  
Table 1 Experimental data

数据集	数据名称	数据量/行
训练集	Wang 等 <sup>[7]</sup>	271329
	SIGHAN 2015 (训练)	3437
	总计	274766
测试集	SIGHAN 2015 (测试)	1099

## 2.2 评价标准及模型参数

本文采用准确率、召回率和 F1 值作为模型的评价指标, 这些指标通常用于 CSC (Chinese spell correction) 任务中的评价。表 2 列出模型的主要参数以及对应的数值。

## 2.3 基线模型

在纠错任务中, Vaswani 等<sup>[4]</sup>提出的 Transformer 效果超越传统的统计模型和 RNN 模型(如混淆集约束下的指针生成网络模型(Confusionset)<sup>[11]</sup>和基于 N-gram 的纠错模型(LMC)<sup>[12]</sup>), 结果如表 3 所示。在应用于纠错任务的 Transformer 模型中, 本文将 Wang 等<sup>[13]</sup>的方法作为基线模型。该模型采用

表 2 模型参数  
Table 2 Model parameters

参数	值	参数	值
layers	6	Heads	8
Hidden-size	512	FF-size	2048
Lr	1.0	Warm-up	8000
Label-smoothing	0.1	Dropout	0.1

表 3 不同改进对模型性能的影响(%)  
Table 3 Impact of different improvements on model performance (%)

模型	SIGHAN 2015			
	P	R	F1	$\Delta F1$
LMC (SIGHAN 2015)	67.60	31.80	43.20	-
Confusionset (ACL 2019)	71.50	59.50	64.90	-
Transformer	71.96	62.57	66.94	-
+DR (baseline) (CCL 2019)	70.10	65.17	67.28	-
+ LF	73.13	62.94	67.65	+0.37
+ ALF	75.14	61.94	67.90	+0.72
+ ON_LSTM	73.88	63.31	68.19	+0.91
+ LF + ON_LSTM	74.77	62.92	68.33	+1.05
+ ALF + ON_LSTM	74.01	62.95	68.03	+0.75
+ LF(DS) + ON_LSTM	74.72	62.46	68.32	+1.04
+ ALF(DS) + ON_LSTM	75.09	63.42	68.76	+1.48

transformer 结构, 并采用语法增强架构来提升模型效果, 我们称之为 DR 模型。DR 模型在 Transformer 的基础上添加动态残差结构, 不仅帮助模型获取更加丰富的语义信息, 还可以有效地减少深层模型因模型过深导致的梯度消失问题。

## 2.4 实验分析

本文共进行 3 组实验: 第一组对比不同模型的效果, 第二组验证局部关注模型以及强化局部关注模型的有效性, 第三组证明 ON\_LSTM 的语法结构信息可以明显地提升模型的准确率和召回率等指标。

对 Transformer 模型的改进如表 3 所示, 共有 3 个独立模型以及一个融合模型, DR 模型为基线模型。LF 模型和 ALF 模型在 Transformer 的自注意力结构中添加高斯偏置, 其 F1 值比基线模型分别提高 0.37% 和 0.72%。ALF 模型的准确率比基线模型有明显的提高, 并且 F1 值比 LF 模型提高 0.35%。

在 Transformer 模型中单独添加 ON\_LSTM 结构使模型的 F1 值提升 0.91%, 证明了该结构的有效性。为了将 Transformer 模型和 ON\_LSTM 结构的优势有效地融合在一起, 采用如图 3 所示的模型结构。值得注意的是, 融合实验的前两组(LF+ON\_LSTM 和 ALF+ON\_LSTM)仅在编码的自注意力部分添加 LF/ALF 结构, 而后两组(LF(DS)+ON\_LSTM 和 ALF(DS)+ON\_LSTM)在编码和解码部分都采用 LF/ALF 结构。从表 3 可以看出, 编码与解码同时采用 ALF 结构可以使模型达到最优的效果。

如表 3 所示, 与 LF 模型相比, ALF 模型准确率的提升幅度比较大, 召回率略微下降, 但 F1 值提升 0.35%。原因是 ALF 模型减少了对错误字符本身的关注, 在错误字符与正确字符之间对应关系的计算方面被弱化, 但加强了对周边范围内字符信息的关注, 会促使模型加强对错误字符的检测能力。模型训练结果如图 4 所示, ALF 模型在准确率提升幅度较大, 召回率与 LF 持平, F1 相对稳定, 但高于 LF 模型。为更好地对比 LF 模型与 ALF 模型的纠错效果, 此部分实验没有添加如图 3 所示的额外结构, 仅在 Transformer 编码与解码的第一层添加 LF 和 ALF。

Transformer 模型中共有三部分使用注意力模块, 本文分别在 Transformer 编码端的自注意力部分、解码端的前向注意力和自注意力部分添加 LF

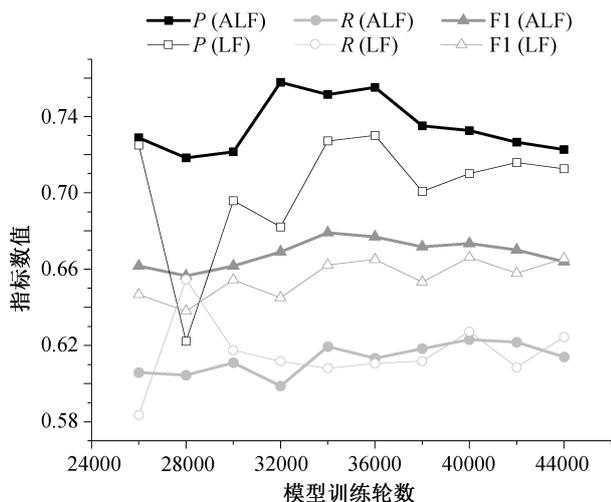


图4 ALF和LF模型的准确率、召回率和F1值  
Fig. 4 Accuracy, recall and F1 of ALF, LF model

结构, 并进行多组实验。从表4可以看出, 在编码和解码部分的自注意力模块添加LF结构会使模型的F1值达到最高。

为探究式(9)中参数  $a$  对ALF模型的影响, 分别设置  $a=1, 2, 4, 0.1$ , 结果见表5。以ES\_DS\_LF作为参考, ALF其他参数与LF保持一致, 则当  $a=0.1$  时, 模型的准确率最高, F1值同样达到最高。

从表3可以看出, 添加X+ON\_LSTM结构模型的准确率有明显的提升, 验证了之前的猜想: 纠错任务中, 语法结构是敏感的, 充分利用语法信息可以帮助模型提升准确率和召回率等指标。

为了验证LSTM结构和ON\_LSTM结构对整体

表4 LF模型添加位置对于模型的影响(%)

Table 4 Influence of LF model adding position on model (%)

模型	P	R	F1
+DSR_LF	74.71	60.64	66.95
+ES_DSR_DS_LF	72.26	62.53	67.04
+ES_DS_LF	73.13	62.94	67.65

说明: ES, DS和DSR分别代表编码层自注意力、解码层自注意力和解码层中对编码层的注意力。

表5 参数  $a$  对模型的影响(%)

Table 5 Effects of parameter  $a$  on the model (%)

模型	P	R	F1
+ES_DS_LF	73.13	62.94	67.65
$a=1$	72.12	61.88	66.61
$a=2$	72.40	63.48	67.65
$a=4$	72.39	60.36	65.83
$a=0.1$	75.14	61.94	67.90

表6 LSTM和ON\_LSTM结构的对比(%)

Table 6 Comparison results of LSTM and ON\_LSTM structures (%)

模型	P	R	F1
+LSTM	71.37	61.41	66.02
+ON_LSTM	71.53	59.90	65.20
+X+LSTM	75.28	61.18	67.50
+X+ON_LSTM	73.88	63.31	68.19

模型的影响, 进行第三组实验, 结果如表6所示。若直接将两种结构的输出结果传入Transformer模型, 纠错的准确率和召回率都有所下降, 但将两种结构的输出结果与输入信息相加后再传入Transformer模型, 纠错效果有很大的提升, 准确率的提升尤为明显。这是因为, 无论是LSTM层还是ON\_LSTM层输出的信息直接传入Transformer, 都会导致极大的信息丢失, X+LSTM(X)/ON\_LSTM(X)结构在输入Transformer模型之前, 在输入数据中获取了文本的其他特征, 与原文的信息融合之后, 对模型起到助推作用。因此, 该结构是对Transformer的一种信息补充, 无论是LSTM的时序关系结构, 还是ON\_LSTM的语法结构, 都能有效地提高Transformer模型的纠错效果, 而语法结构信息对Transformer模型的纠错效果帮助最大。

在将不同模型进行融合的过程中, 我们发现, 如果将ON\_LSTM的结果直接输入带有高斯偏置的编码层中, 最终得到的结果性能会下降。因此, 语法结构信息会对LF/ALF产生影响。为了解决这个问题, 本文将输入的信息分为两个部分, 一部分通过ON\_LSTM层直接进入Transformer的编码层, 另外一部分输入信息则送入带有高斯偏置的编码层中, 并将两部分结构的输出结果相加后送入编码层。实验证明, 这种方式可以优化模型结构, 加速收敛, 提升模型纠错率。

Transformer对由字错误或词错误导致的句子的语法错误相对不敏感且纠错不准确, 而本文模型对此类型错误纠正的正表现较好, 并且错误纠正的范围更大(图5)。

### 3 结论

本文从两个方面对Transformer模型进行改进。首先采用局部关注的思想, 在自注意力部分添加高斯分布偏置项, 用于提高模型对局部信息的关

错误语句:	我打算要去法国流行,我想你要不要跟我去,有朋友一起去比较好。
Transformer:	我打算要去法国流行,我想你要不要跟我去,有朋友一起去比较好。
本文:	我打算要去法国旅行,我想你要不要跟我去,有朋友一起去比较好。
错误语句:	他们最喜欢的地方是高雄的柴山,那里很漂亮友山友海。
Transformer:	他们最喜欢的地方是高雄的柴山,那里很漂亮友山有海。
本文:	他们最喜欢的地方是高雄的柴山,那里很漂亮有山有海。
错误语句:	他正要喝点水,老师就进教师来了。
Transformer:	他正要喝点水,老师就尽教室来了。
本文:	他正要喝点水,老师就进教室来了。

图 5 Transformer 以及本文模型在不同错误语句上的表现

Fig. 5 Transformer and our model's performance on different error statements

注,然后针对拼写纠错任务的语法结构特点,采用 ON\_LSTM 结构,加强模型对语法结构信息的获取。实验数据表明,两种方法都会帮助模型提高纠错效果,且二者的融合结构使模型提升更多。

中文拼写纠错是一个依赖知识的任务,知识是文本信息构成的知识库。在对错误字符进行处理时,需要大量的先验知识,所以将来的工作就是构建文本知识库,用于辅助模型对错误文本进行检测和修改。

### 参考文献

- [1] Kukich K. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 1992, 24(4): 377–439
- [2] Huang Q, Huang P, Zhang X, et al. Chinese spelling check system based on tri-gram model // *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*. Wuhan, 2014: 173–178
- [3] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks // *Advances in Neural Information Processing Systems*. Montreal, 2014: 3104–3112
- [4] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need // *Neural Information Processing Systems*. Red Hook, 2017: 5998–6008
- [5] Yang B, Tu Z, Wong D F, et al. Modeling localness for self-attention networks // *Empirical Methods in Natural Language Processing*. Brussels, 2018: 4449–4458
- [6] Chollampatt S, Ng H T. A multilayer convolutional encoder-decoder neural network for grammatical error correction [EB/OL]. (2018–01–26)[2020–05–01]. <https://arxiv.org/abs/1801.08831>
- [7] Wang D, Song Y, Li J, et al. A hybrid approach to automatic corpus generation for Chinese spelling check // *Empirical Methods in natural language processing*. Brussels, 2018: 2517–2527
- [8] Shen Y, Tan S, Sordoni A, et al. Ordered neurons: integrating tree structures into recurrent neural networks [EB/OL]. (2019–05–08)[2020–05–01]. <https://arxiv.org/abs/1810.09536>
- [9] Anastasopoulos A, Chiang D. Tied multitask learning for neural speech translation [EB/OL]. (2018–04–26)[2020–05–01]. <https://arxiv.org/abs/1802.06655>
- [10] Tseng Y H, Lee L H, Chang L P, et al. Introduction to SIGHAN 2015 bake-off for Chinese spelling check // *Proceedings of the 8th SIGHAN Workshop on Chinese Language Processing (SIGHAN'15)*. Beijing, 2015: 32–37
- [11] Wang D, Tay Y, Zhong L, et al. Confusionset-guided pointer networks for Chinese spelling check // *Meeting of the Association for Computational Linguistics*. Florence, 2019: 5780–5785
- [12] Xie W, Huang P, Zhang X, et al. Chinese spelling check system based on N-gram model // *Proceedings of the Eighth SIGHAN Workshop on Chinese Language*. Beijing, 2015: 128–136
- [13] Wang Chencheng, Yang Liner, Wang Yingying, et al. 基于 Transformer 增强架构的中文语法纠错方法. *中文信息学报*, 2020, 34(6): 106–114