基于BERT和多特征融合嵌入的中文拼写检查

刘哲，殷成凤，李天瑞

引用本文

刘哲，殷成凤，李天瑞.基于BERT和多特征融合嵌入的中文拼写检查[J]．计算机科学, 2023, 50(3): 282-290.

LIU Zhe, YIN Chengfeng, LI Tianrui. Chinese Spelling Check Based on BERT and Multi-feature Fusion Embedding [J]. Computer Science, 2023, 50(3): 282-290.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220100104

基于 BERT 和多特征融合嵌入的中文拼写检查

刘　哲[1]　殷成凤[1]　李天瑞[1,2]
1 西南交通大学计算机与人工智能学院　成都 611756
2 综合交通大数据应用技术国家工程实验室　成都 611756
(liuzhe@my.swjtu.edu.cn)

摘　要　由于汉字的多样性和中文语义表达的复杂性,中文拼写检查仍是一项重要且富有挑战性的任务。现有的解决方法通常存在无法深入挖掘文本语义的问题,且在利用汉字独特的相似性特征时往往通过预先建立的外部资源或是启发式规则来学习错误字符与正确字符之间的映射关系。文中提出了一种融合汉字多特征嵌入的端到端中文拼写检查算法模型 BFMBERT(BiGRU-Fusion Mask BERT)。该模型首先利用结合混淆集的预训练任务使 BERT 学习中文拼写错误知识,然后使用双向 GRU 网络捕获文本中每个字符错误的概率,利用该概率计算汉字语义、拼音和字形特征的融合嵌入表示,最后将这种融合嵌入输入到 BERT 中的掩码语言模型(Mask Language Model,MLM)以预测正确字符。在 SIGHAN 2015 基准数据集上对 BFMBERT 进行了评测,取得了 82.2 的 F1 值,其性能优于其他基线模型。

关键词:中文拼写检查;BERT;文本校对;掩码语言模型;字词错误校对;预训练模型
中图法分类号　TP181

Chinese Spelling Check Based on BERT and Multi-feature Fusion Embedding

LIU Zhe[1],YIN Chengfeng[1] and LI Tianrui[1,2]
1 School of Computing and Artificial Intelligence,Southwest Jiaotong University,Chengdu 611756,China
2 National Engineering Laboratory of Integrated Transportation Big Data Application Technology,Chengdu 611756,China

Abstract　Due to the diversity of Chinese characters and the complexity of Chinese semantic expressions,Chinese spelling checking is still an important and challenging task. Existing solutions usually suffer from the inability to dig deeper into the text semantics and often learn the mapping relationship between incorrect and correct characters through pre-established external resources or heuristic rules when exploiting the unique similarity features of Chinese characters. This paper proposes an end-to-end Chinese spelling checking algorithm model BFMBERT(BiGRU-Fusion Mask BERT) that incorporates multi-feature embedding of Chinese characters. The model first uses a pre-training task combining confusion sets to make BERT learn Chinese spelling error knowledge. It then employs a bi-directional GRU network to capture the probability of error for each character in the text. Furthermore,it applies this probability to compute a fusion embedding incorporating semantic,pinyin,and glyph features of Chinese characters. Finally,it feeds this fusion embedding into a mask language model in BERT to predict correct characters. BFMBERT is evaluated on the SIGHAN 2015 benchmark dataset and achieves an F1 value of 82.2,outperforming other baseline models.

Keywords　Chinese spelling check,BERT,Text proofreading,Masked language model,Word error proofreading,Pre-training model

1　引言

随着互联网技术的飞速发展,电子文本的数量呈指数增长。尽管海量文本数据可以使我们更轻松地获取相关信息,但是网络数据的爆炸性增长也导致互联网文本信息质量的大幅下降,如何检测与纠正中文文本的字词错误是一个热门的研究领域。

中文拼写检查(Chinese Spelling Check,CSC)是一项旨在检测并纠正中文文本中字词错用的任务[1-3]。不同于英文,中文是一种表意文字,且不会出现汉字库中没有的错字,只存在别字,这使得中文拼写检查的难度远大于英文。对于错误检测,中文字词之间并没有天然分隔符,很难判断句子中是否有字词错误以及哪些字词存在拼写错误[4]。并且含有字词错误的句子并不能很好地分词,因此以词为单位进行中文拼写

到稿日期:2022-01-11　返修日期:2022-08-25
基金项目:国家自然科学基金(61773324);四川省重点研发项目(2020YFG0035);中央高校基本科研业务费专项资金(2682021ZTPY097)
This work was supported by the the National Natural Science Foundation of China(61773324),Sichuan Key R&D Project(2020YFG0035) and Fundamental Research Funds for the Central Universities of Ministry of Education of China(2682021ZTPY097).
通信作者:李天瑞(trli@swjtu.edu.cn)

检查并不合适。此外，很多拼写错误发生在单个字词之间（如“人”和“入”），必须结合上下文语义信息才能准确地识别出错误。对于错误纠正，如何从多达 10 000 个常见汉字候选中选择正确的字符也非常困难[5]。

据研究[6]，汉字字词错误大约 83% 与语音相似性有关，48% 与视觉相似性有关。中文拼写错误示例如表 1 所列，错字大多数为音近字或形近字。在错误纠正阶段，过去的研究通常凭借一个大规模的由相近字符对组成的混淆集（Confusion Set）来限制候选字符的范围。但构建混淆集非常复杂，且其很难覆盖所有的候选字符。除此之外，语言模型[7-8]也是被大规模使用的方法，往往被用来进行错误检测和评估纠正后的句子是否通顺，但其检测和定位错误位置的效率和准确性并不高，且纠正后句子是否通顺的阈值分数需凭经验来取值，这削弱了它的有效性。

表 1　中文拼写错误示例

Table 1　Examples of Chinese spelling errors

类型	句子	纠正后的字
音近字	他在热身示(shi)出了意外	时(shi)
形近字	他驳斥这顶(ding)报导	项(xiang)

近几年，基于 BERT[9] 的预训练模型出现后，CSC 任务有了长足发展。BERT 有着强大的语义能力，能够利用掩码语言模型结合上下文语义预测句子中每个位置最可能正确的字。但在检测阶段，BERT 只能将所有字符视为错误字符，这导致检测准确率不高。在纠正阶段，BERT 只能基于上下文语义来进行预测，并不能很好地利用汉字独特的字形和字音特征。

因此，本文提出了一种结合 BERT 和汉字多特征融合嵌入的网络模型 BFMBERT，其包含两个网络：基于双向门控循环网络（Bidirectional Gating Recurrent Unit，BiGRU）的检测网络和基于 BERT 的纠正网络。检测网络预测句子中每个位置字符错误的概率，然后基于错误概率计算融合了语义、字形与拼音特征的字符嵌入表示来替代混淆集。当字符的错误概率越高时，融合嵌入越接近该字的拼音与字形的融合嵌入，反之则退化为字符本身的语义嵌入。这与人类进行中文拼写检查时类似。当我们认为这个字符错误时，会根据上下文语义去寻找错误字符的音近字或形近字来进行纠正；当我们认为该字符正确时，则不需要考虑其相近字符。最后将该融合嵌入输入至纠正网络以预测正确的字符。在 SIGHAN 2015 数据集上的实验结果验证了 BFMBERT 的有效性。本文的主要工作如下：

（1）为了弥补传统的混淆集无法覆盖所有的相近字符以及不灵活的缺点，本文提出了一种新的融合嵌入，能将中文字符的语义、拼音和字形特征融合到语义空间中。其中，语义嵌入提取自 BERT 嵌入，字形嵌入从不同字体的字符的视觉表面形式中捕获字符的语义，拼音嵌入则能建模同一字符不同读音时的不同语义，从而获得了单个字符的交叉语素。将语义嵌入、拼音嵌入和字形嵌入融合形成融合嵌入，能够向量化表示汉字独特的语义、字形和拼音特征信息。

（2）BERT 在掩码预训练任务中将字符替换为固定的掩码标记“[MASK]”，而该标记在下游任务中并不存在。为了缩小 BERT 预训练任务和中文拼写检查任务的差异，本文提出了一种新的预训练掩码策略。该策略不使用固定的掩码标记，而是在混淆集中选择与被替换字符相近的随机字符进行替换，以此模拟从错误字符纠正为正确字符的过程。相关研究[10]和实验表明，这种预训练策略能够使得 BERT 在预训练阶段学习语义和拼写的错误知识。

（3）由于 BERT 在进行中文拼写检查时只能将所有字符视为错误字符，导致误纠率较高，错误检测能力较弱。本文提出了一种端到端的 CSC 模型 BFMBERT，该模型采用联合训练检测网络和纠正网络的方式将汉字字义、拼音和字形特征融入到语言表示中，在一个统一的框架下优化检测和纠正网络。实验结果表明，与基线模型相比，BFMBERT 在错误检测和错误纠正上均取得了显著提升。

2　相关工作

以往将中文拼写检查的研究主要分为 3 类：基于规则的方法、基于统计的方法和基于深度学习的方法。Mangu 等[11]提出了一种从一小部分易于理解的规则中自动获取语言知识的方法。Chang 等[12]通过融合基于规则的方法分别处理字错误和词错误。Jiang 等[13]提出了一种新的语法规则体系，用于解决拼写错误和语法错误。Xiong 等[14]提出的 HANSpeller 是一个统一的中文拼写纠错框架，其融合了基于扩展的隐马尔可夫（Hidden Markov Model，HMM）模型、基于排序的模型和基于规则的模型。基于规则的方法比较浅显易懂，但是这些规则很难覆盖所有的中文拼写错误，这导致这类方法往往不具有泛化性。并且，若想尽可能覆盖更多的错误情况，则需要耗费大量精力去构建更多规则。

在基于统计的方法中，研究者通常采用错误检测、错误纠正候选生成和候选排序这 3 个步骤的策略[3,8,15-19]。Chiu 等[20]使用噪声信道模型来进行错误纠正。Wang 等[21]使用词向量和基于条件随机场（Conditional Random Field，CRF）的错误检测器，来寻找可能的拼写错误并提供更正建议。*N*-Gram 语言模型也被大规模用于候选排序以及评估纠正后句子的质量。Huang 等[22]使用基于字符的三元组语言模型来确定在检测和纠正可能错误后的最流畅的语句。Liu 等[23]使用 *N*-Gram 模型，通过分词来检测错误，并将其与错误纠正的启发式规则相结合。基于统计的方法遵循流水线模式，导致其可能出现错误传递。这类方法往往采用经验阈值来判断句子是否通顺，这也削弱了模型的性能和灵活性。此外，基于统计的方法也无法深入地挖掘语义信息，容易导致误纠或漏纠。

深度学习在包括中文拼写检查的许多自然语言处理（Natural Language Process，NLP）任务上取得了优异的成绩。Wang 等[24]提出了一种序列到序列（Sequence To Sequence，Seq2Seq）的指针网络，该网络利用复制机制从混淆集中生成汉字。Tao 等[25]结合长短期记忆网络和模型集成方法，提高了中文拼写检查的准确性。Yang 等[26]提出利用 Spark Streaming 流式框架并行处理中文文本校对，可以加快处理中文拼写检查的速度。Li 等[27]利用神经机器翻译将错误句子“翻译”为正确句子。虽然基于自回归的 Seq2Seq 模型具有纠正拼写错误的能力，但是其推理速度较慢，而且对于中文拼写检查任务，输入和输出通常非常接近，完全生成整个句子非常“浪费”[28]。近几年，基于掩码语言模型的预训练模型，如

BERT[9]，RoBERTa[29]，XLNET[30]，ELECTRA[31]，在许多NLP任务中取得了巨大成功。因为输入和输出序列的长度相同，并且正确和错误汉字的位置对应，所以使用此类非自回归语言模型处理中文拼写检查任务更具优势[32]。Hong[33]提出的FASPell抛弃了传统的混淆集，采用预训练的MLM作为编码器，同时训练以置信度和字符字形字音相似度为基础的解码器。Cheng等[34]在预训练阶段通过图卷积网络融合字符之间的相似性知识。这些方法将字符相似性信息作为外部知识，但是离散的候选字符选择阻碍了BERT通过反向传播直接进行学习。Zhang等[35]提出了一种软屏蔽的BERT模型，该模型首先预测每个单词的拼写错误概率，然后使用这些概率计算软蒙蔽的字符嵌入以进行更正，但是他们没有使用任何汉字相似性知识。Huang等[36]从多模态信息中提取汉字的拼音和字形表示，通过自适应门控机制将其融入到预训练模型中，但其提取汉字特征的方法较为困难。

3 拼写检查模型

给定一个长度为 n 的句子 $X=\{x_1,x_2,\cdots,x_n\}$，中文拼写检查任务的目标是在字符级别上检测其中的拼写错误，并输出其对应的正确句子 $Y=\{y_1,y_2,\cdots,y_n\}$。

本文提出的端到端拼写检查模型BFMBERT如图1所示，该模型主要包含检测网络和纠正网络两部分。检测网络接收 X 的BERT嵌入作为输入，并预测每个字符的错误概率 $P=\{p_1,p_2,\cdots,p_n\}$。纠正网络以BERT模型为基础，先以错误概率为权重，计算 X 的BERT嵌入和拼音字形融合嵌入 $X_f=\{x_{f1},x_{f2},\cdots,x_{fn}\}$ 的加权和并将其作为输入，在经过BERT编码层后得到表示 h^c，再通过softmax层计算在BERT字典上的预测分布，最终得到纠正后的句子 Y。

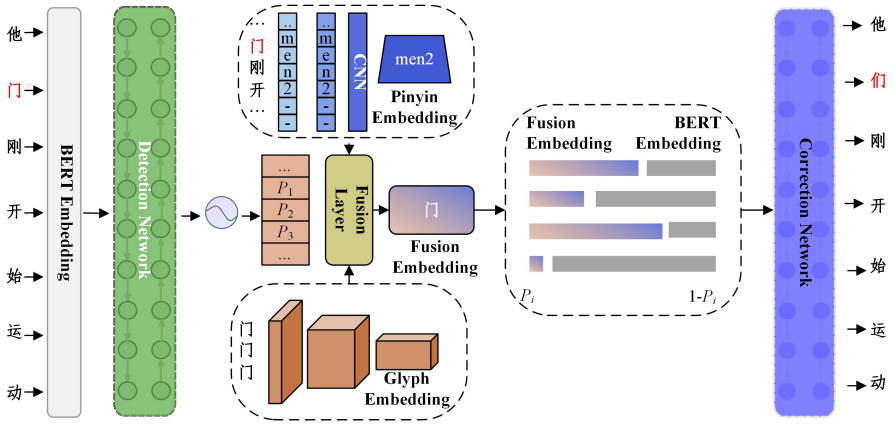


图1 BFMBERT模型的结构

Fig. 1 Model structure of BFMBERT

3.1 基于混淆集的预训练掩码策略

“预训练-微调”范式[9]已在文本分类、情感分析等NLP任务上被证明有效。但是，这些任务的输入与预训练模型的输入具有相同的分布，而CSC任务的输入句子存在错误，这导致其与预训练模型的输入样本分布不同。除此之外，在掩码任务中BERT将掩码标记固定为“[MASK]”，这个标记在CSC任务中并不存在。因此，本文提出了一种新的针对CSC任务的掩码策略，该策略并未使用“[MASK]”作为唯一标记，还增加了随机相似字符替换的方法，以此减小预训练任务与错误纠正任务的差距。同时，该策略采用了动态掩码[29]的方法，每次训练随机掩盖25%不同的字符，并放弃了与中文拼写检查无关的下一句预测任务(Next Sentence Prediction，NSP)。表2列出了这种掩码策略的样例，具体的掩码策略如下。

(1)70%的字符被替换为混淆集中的相似字符。混淆集是包含汉字及其相似字的映射集合，本文会从被替换字符的相似字中随机挑选汉字进行替换。如果混淆集中没有被选择的待替换字符，本文则会将其保留为“[MASK]”标记，这帮助了BERT学会纠正错字。

(2)为了防止BERT倾向于所有输入都应该被纠正，15%的字符不进行改变。

(3)混淆集中的相似字符是有限的，但是在真实的错误文本中可能存在任何错用的汉字导致的拼写错误。因此，为了提高泛化能力，15%的字符被替换为随机字符。

表2 基于混淆集的预训练掩码策略样例

Table 2 Example of pre-training mask strategy based on confusion set

掩码类型	掩码后的句子
原始语句	他们刚开始运动
BERT掩码	他[MASK]刚开始运动
相似字掩码	他门刚开始运动
随机掩码	他看刚开始运动
不变	他们刚开始运动

3.2 检测网络

如图2所示，检测网络是一个序列标注模型。其输入是序列 X 的BERT嵌入 $E=\{e_1,e_2,\cdots,e_n\}$。BERT嵌入是序列 X 字符嵌入、位置嵌入和句子嵌入的总和。输出是一个标签序列 $T=\{t_1,t_2,\cdots,t_n\}$，其中 t_i 代表字符 x_i 的标签，本文分别用“1”和“0”来标记拼写错误和正确的字符。

本文采用双向门控循环网络来实现该模型。门控循环网络(Gating Recurrent Unit，GRU)由长短期记忆网络(Long Short Term Memory，LSTM)修改而来，具体来说GRU使用更新门来替代LSTM中的遗忘门和输入门，并将内部状态(Cell State)和隐藏状态进行合并。相比LSTM，GRU结构更加简单，训练更加高效。本文中，其隐藏状态被定义为式(1)：

$$(\overrightarrow{h_t^d})=GRU(\overrightarrow{h_{(i-1)}^d},e_i) \tag{1}$$

其中，GRU 表示GRU函数，e_i 是字符 x_i 的嵌入，$\overrightarrow{h_i^d}$ 表示检测网络此刻的隐藏状态，$\overrightarrow{h_{(i-1)}^d}$ 表示上一刻的隐藏状态。

单向的 GRU 按文本序列的顺序接收输入，其只能处理前文信息。而双向门控循环网络结合正向 GRU 网络和逆向 GRU 网络对输入序列提取双向语义特征信息，如式(2)和式(3)所示，其隐藏状态被定义为：

$$(\overleftarrow{h_t^d}) = GRU(\overleftarrow{h_{(i-1)}^d}, e_i) \tag{2}$$

$$h_t^d = [\overrightarrow{h_t^d}, \overleftarrow{h_t^d}] \tag{3}$$

其中，$[\overrightarrow{h_t^d}, \overleftarrow{h_t^d}]$代表两个方向的 GRU 网络隐藏状态的串联。对于输入序列中的每个字符存在概率 p_i，p_i 越高代表字符 x_i 的错误可能性越大，在本文提出的检测网络中 p_i 被定义为式(4)：

$$p_i = P_d(t_i = 1 | X) = \delta(W_d h_i^d + b_d) \tag{4}$$

其中，$P_d(t_i = 1 | X)$ 表示检测网络给出的条件概率，δ 表示 sigmod 函数，h_i^d 是 BiGRU 最后一层的隐藏状态，W_d 和 b_d 是检测网络的参数。

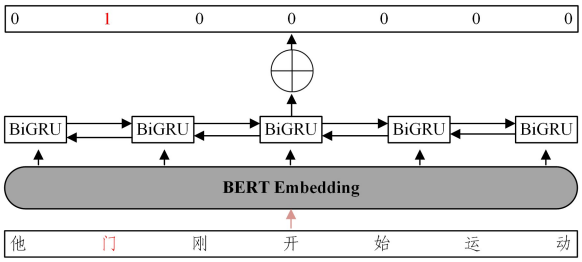


图 2　检测网络

Fig. 2　Detection network

3.3　多特征融合嵌入

3.3.1　拼音嵌入

汉字存在许多音近字甚至同音字，这是导致拼写错误的主要原因，因此如何提取汉字的拼音特征至关重要。本文使用拼音嵌入来对字符的读音进行建模。首先通过开源工具包 pypinyin 获取句子的拼音序列，汉语字符的拼音是一串罗马尼亚字符序列，且带有 5 种音调。本文使用 5 种固定标记代表 5 种音调，并将其添加到拼音序列的末尾，例如“门”的拼音序列为“men2”。本文将输入的拼音序列长度固定为 8，当拼音序列的实际长度不足 8 时，会添加特殊符号“—”。

如图 3 所示，模型先将拼音序列输入到核心大小为 2 的卷积神经网络(Convolutional Neural Networks，CNN)中，采用最大池推导后得到拼音嵌入，使得输出的向量维度不受输入拼音序列长度的影响。

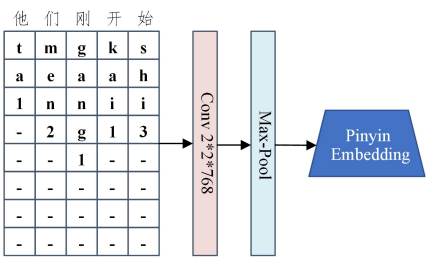


图 3　拼音嵌入

Fig. 3　Pinyin embedding

3.3.2　字形嵌入

受 Meng 等[37]提出的 Glyce 的启发，本文从汉字的字体中提取汉字的字形特征。字体文件包含汉字的位图信息，提供了天然的汉字视觉特征。本文采用了仿宋、行楷和楷书 3 种中文字体，并将每种字体实例化成大小为 24 * 24 的图像，每幅图像的浮点像素范围为 0～255。如式(5)所示，与 Glyce 不同，本文没有采用 CNN 提取图像特征，而是首先将 3 种字体中获取的 24 * 24 * 3 维度的矢量通过嵌入层转换为一维，然后将压平后的矢量输入全连接层得到字形嵌入。

$$e_g = FC(e_{g1} + e_{g2} + e_{g3}) \tag{5}$$

其中，e_g 表示字形嵌入，FC 是全连接层，e_{gi} 分别代表从不同字体获取的汉字特征。

3.3.3　融合拼音嵌入和字形嵌入

本文提出了 3 种不同的融合拼音嵌入和字形嵌入的方法，并在后文进行了对比实验。

(1)直接将两种特征拼接是最直观的方法之一，如式(6)所示。本文将拼音嵌入和字形嵌入连接起来，然后在后续的网络层对该操作进行自适应训练。

$$e_f = \text{concat}(e_p, e_g) \tag{6}$$

(2)另一种直观的方法是将两种特征相加，如式(7)所示。

$$e_f = e_p + e_g \tag{7}$$

(3)如式(8)所示，第三种方法是在拼接后添加一个线性层以帮助其融合，并控制拼接后的维度，这种方法节省了空间并提高了模型的训练速度。

$$e_f = linear(concat(e_p, e_g)) \tag{8}$$

其中，$e_f \in \mathbb{R}^{d_f}$ 是融合拼音与字形特征的融合嵌入，d_f 是融合嵌入维度，$e_p \in \mathbb{R}^{d_p}$ 是拼音嵌入，d_p 是拼音嵌入维度；$e_g \in \mathbb{R}^{d_g}$ 是字形嵌入，d_g 是字形嵌入维度，$Linear$ 是线性层。当使用第一种融合方式时，拼音嵌入维度 d_p 和字形嵌入维度 d_g 为其他融合方式的一半，使不同融合方式得到的拼音字形融合嵌入维度 d_f 保持一致。

3.3.4　基于错误概率融合语义嵌入和拼音字形嵌入

为了平衡语义特征(BERT 嵌入)和拼音字形融合特征(拼音字形融合嵌入)的重要性，本文使用检测网络输出的拼写错误概率为权重，通过线性组合将 BERT 嵌入和拼音字形嵌入结合，如式(9)所示：

$$e_{mi} = (1 - p_i) * e_i + p_i * e_{fi} \tag{9}$$

其中，$e_{mi} \in \mathbb{R}^{d_m}$ 是融合了拼音字形特征和语义特征的融合嵌入，d_m 是融合了 3 种特征嵌入的维度，p_i 是字符的拼写错误概率，e_i 是字符的 BERT 嵌入，e_{fi} 是字符的拼音字形融合嵌入。当 $p_i = 0$ 时，即拼写正确的情况下，模型直接使用其字符嵌入 e_i；当 $p_i = 1$ 时，即拼写错误的情况下，模型使用其拼音字形融合嵌入寻找相近字。

3.4　纠正网络

如图 4 所示，纠正网络是基于 BERT 的多分类模型。其输入是融合了拼音字形特征和语义特征的融合嵌入 $E_m = \{e_{m1}, e_{m2}, \cdots, e_{mn}\}$，输出是正确的文本序列 $Y = \{y_1, y_1, \cdots, y_n\}$。BERT 由 12 层 Transformer[38]编码网络组成，编码网络的核心是自注意力模块，如式(10)所示：

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{10}$$

其中，$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ 均是输入的向量矩阵，$\boldsymbol{K}^{\mathrm{T}}$ 是 $\boldsymbol{K}$ 的转置矩阵，d_k 是输入向量的维度。自注意力模块的目标是，计算句子中的

每个字符对句子中其他字符的关联性以及句子中不同字符的重要程度。利用这种相互关系和权重来获得每个字符新的语义特征表示，使得其不仅包含该字符本身的特征，还蕴含了与其他字符的关系，因此 Transformer 编码网络具有更深更全面的语义信息。

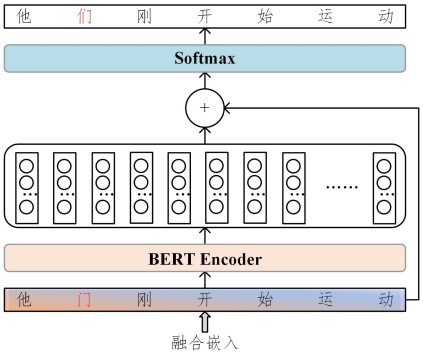


图 4 纠正网络

Fig. 4 Correction network

同时，为了扩展模型在句子中不同位置提取特征的能力，扩大注意力模块的表示空间，如式(11)－式(13)所示，Transformer 采用了多头自注意力模块和前馈神经网络。

$$MultiHead(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=Concat(head_1,\cdots,head_h)W^o \tag{11}$$

$$head_i=Attention(\boldsymbol{Q}W_i^Q,\boldsymbol{K}W_i^k,\boldsymbol{V}W_i^v) \tag{12}$$

$$FFN(X)=\max(0,XW_1+b_1)W_2+b_2 \tag{13}$$

其中，$MultiHead$ 代表多头自注意力，$Attention$ 表示自注意力，FNN 表示前馈神经网络，W^o，W_i^Q，W_i^K，W_i^V，W_1，W_2，b_1 和 b_2 均是函数的参数。

在纠正网络中，每个字符的融合嵌入后会被传递到 BERT 编码器层，用于计算与 BERT 字典中所有字符嵌入的相似度，以获得在词汇表上的预测分布，其在最后一层的隐藏状态定义为式(14)：

$$h_i^c=BERTEncoder(e_{mi}) \tag{14}$$

为了防止梯度消失，纠正网络不会直接使用 BERT 最后一层的隐藏状态，而是与字符 x_i 的 BERT 嵌入 e_i 残差连接线性组合后得到新的隐藏状态，如式(15)所示：

$$h_i'=h_i^c+e_i \tag{15}$$

最后，如式(16)所示，利用 softmax 函数计算字典中每个字符为正确字符的概率，选择概率最高的字符作为纠正后的字符。

$$P_c(y_i=j|X)=\text{softmax}(Wh_i'+b)[j] \tag{16}$$

其中，$P_c(y_i=j|X)$是字符 x_i 被纠正为 j 的条件概率，h_i'是添加了残差链接的隐藏状态，W 和 b 是纠正网络的参数。

3.5 模型训练

如式(17)－式(19)所示，BFMBERT 的训练过程由两个目标驱动，分别是检测网络损失函数和纠正网络损失函数，总的目标由两个损失函数线性组合构成。

$$\mathcal{L}_d=-\sum_{i=1}^{n}\log p_d(t_i|X;\theta_d) \tag{17}$$

$$\mathcal{L}_c=-\sum_{i=1}^{n}\log p_c(y_i|X;\theta_c) \tag{18}$$

$$\mathcal{L}=\lambda\cdot\mathcal{L}_c+(1-\lambda)\cdot\mathcal{L}_d \tag{19}$$

其中，$\mathcal{L}_d$ 和 $\mathcal{L}_c$ 分别代表检测网络和纠正网络的损失函数，θ_d 和 θ_c 是检测网络和纠正网络的参数，$\mathcal{L}$是整个模型联合训练的损失函数，λ 是损失函数线性组合的参数。

4 实验

4.1 实验数据和预处理

本文的实验在 SIGHAN 2015[3]基准数据集上进行评估，SIGHAN 评测规定参赛者可以使用任意的语言和计算资源，过去的参赛者和研究使用了往年的 SIGHAN 评测数据集来训练模型，因此本文也同样采用了 SIGHAN 2014[2]的训练集以及 Wang 等[39]自动生成的大型语料库作为训练数据。同时，在预训练阶段使用已经转化为简体中文的维基百科语料。

因为 SIGHAN 数据集来源于外国学生书写的繁体中文，本文通过 OpenCC 将它们转换为简体中文，修改了一些转换错误的汉字，并剔除了一些转换后句子长度发生改变的样本。为了提高泛化性，本文随机提取了 1%的自动生成语料来生成无错误的训练数据以增加无错误语句的比例。具体的数据集统计数据如表 3 所列。

表 3 数据集统计

Table 3 Dataset statistics

数据集	语句数	平均长度
自动生成语料库	271385	44.4
SIGHAN2014 训练集	3428	49.6
SIGHAN2015 训练集	2336	31.3
SIGHAN2015 测试集	1100	30.5

4.2 评价指标

本文采用官方提供的工具[2-3]进行评估。该脚本在句级别上分别评估错误检测和错误纠正的性能，这种度量方式比字符级别更为严格，因为只有句子中的所有错误均被检测和纠正后才会被视为正样本。具体的评价指标有准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 值(F1)，其计算式分别如下：

$$Accuracy=\frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{20}$$

$$Precision=\frac{TP}{TP+FP} \tag{21}$$

$$Recall=\frac{TP}{TP+FN} \tag{22}$$

$$F1=\frac{2*Precision*Recall}{(Precision+Recall)} \tag{23}$$

其中，TP 表示正确识别的有拼写错误的样本数，FP 表示识别为有拼写错误但是实际上没有拼写错误的样本数，TN 表示正确识别的没有拼写错误的样本数，FN 表示没有识别的有拼写错误的样本数。

4.3 参数设置

在预训练阶段，模型采用基于全词掩盖的 BERT-WWM-EXT[40]的权重进行初始化。本文只使用了基于 3.2 节提出的混淆集掩码策略的掩码任务，Transformer 编码器的参数与初始参数完全相同，采用了带有 12 个注意力头的 12 层编码器，学习率设置为 5×10^{-5}。

在微调阶段使用 AdamW[41]优化器，模型的隐藏状态向量大小为 768，采用学习率预热策略进行训练，最大学习率为 1×10^{-4}，权重衰减系数为 5×10^{-8}，批量大小为 16，最佳损失

函数权重为 0.2，使用了累计梯度策略。当使用了 3.4.3 节提出的拼接嵌入融合方式时，拼音嵌入 d_p 和字形嵌入 d_g 的维度大小均为 384；使用其他融合方式时，拼音嵌入 d_p 和字形嵌入 d_g 的维度大小均为 768。

4.4　实验结果

为了证明 BFMBERT 的有效性，本文使用以下方法作为基线模型进行比较。

(1)NTOU[3]：采用基于 *N*-gram 模型和规则的分类器进行中文拼写检查。

(2)NCTU-NTUT[21]：利用字级别的嵌入和条件随机场的方法寻找句子中的拼写错误。

(3)HanSpeller[14]：融合了基于隐马尔可夫的模型和基于规则的模型，并对多个纠正结果进行了重排序。

(4)Hybrid[39]：使用序列标注的方法在自动生成的数据集上训练基于 LSTM 的模型。

(5)ConfusionSet[24]：将中文拼写检查视为序列到序列的任务，通过指针网络使用混淆集来缩小解码器候选范围。

(6)CPLM-CSC[42]：使用字符级别预训练模型和序列标记处理中文拼写检查，利用相关阈值进行候选字符的筛选。

(7)FASPell[33]：使用 BERT 作为降噪编码器，并以置信度和字形字音相似度训练解码器。

(8)Soft-Masked BERT[35]：对于句子中的每个标记，线性组合其字符嵌入和“[MASK]”嵌入，利用检测网络微调 BERT 进行错误纠正。

(9)SpellGCN[34]：利用图神经网络建模混淆集中相似字之间的字形和字音特征，并将其结合 BERT 进行预测。

(10)Chunk[43]：利用基于块的统一框架，使用语义候选结合汉字特征扩充混淆集。

(11) BERT-Finetune[44]：直接将语句输入到 BERT-WWM-EXT[38]，在 BERT 最后一层后添加 softmax 层来预测正确字符。

如表 4 所列，本文提出的 BFMBERT 在 SIGHAN2015 数据集上相比基线模型具有更好的错误检测和错误纠正能力。BFMBERT 的性能优于早期的传统机器学习方法和基于规则的方法，传统的方法非常依赖于收集规则和混淆集的建立，这也导致其开发成本更高且丢失了灵活性。

表 4　各模型的实验结果

Table 4　Experimental results of different models

模型	错误检测				错误纠正			
	Acc.	*Prec.*	*Rec.*	*F*1	*Acc.*	*Prec.*	*Rec.*	*F*1
NTOU	42.2	42.2	41.8	42.0	39.0	38.1	35.2	36.6
NCTU-NTUT	60.1	71.7	33.6	45.7	56.4	66.3	26.1	37.5
HanSpeller	70.1	80.3	53.3	64.0	69.2	79.7	51.5	62.5
Hybrid	—	56.6	69.4	62.3	—	—	—	57.1
ConfusionSet	—	66.8	73.1	69.8	—	71.5	59.5	64.9
CPLM-CSC	68.6	70.0	65.3	67.5	67.1	69.0	62.2	65.4
FASPell	74.2	67.6	60.0	63.5	73.7	66.6	59.1	62.6
Soft-Masked BERT	80.9	73.7	73.2	73.5	77.4	66.7	66.2	66.4
SpellGCN	83.7	85.9	80.6	83.1	82.2	85.4	77.6	81.3
Chunk	76.8	**88.1**	62.0	72.8	74.6	**87.3**	57.6	69.4
BERT-Finetune	81.7	85.2	76.0	80.3	80.3	84.7	73.5	78.7
BFMBERT	**84.1**	86.1	**81.3**	**83.6**	**82.9**	85.8	**78.9**	**82.2**

在深度学习方法中，Hybrid 使用 LSTM 模型来进行建模，但其效果不理想，证明 LSTM 模型没有深入挖掘中文拼写错误的能力。ConfusionSet 的结果是字符级别的，通常来说性能更好，但仍然远落后于 BFMBERT，这表明序列到序列模型即便在混淆集帮助下能减小候选范围，也仍然不适合中文拼写检查任务。以 BERT 为基础的深度学习方法总体来说性能更好，这说明 BERT 的语义挖掘能力和 MLM 模型更适用于中文拼写检查任务。与直接使用 BERT 微调相比，BFMBERT 的性能更好，证明了错误检测网络和融合嵌入的有效性。CPLM-CSC 使用人为定义的字形、字音相似度、排名阈值和概率阈值来筛选 BERT 生成的候选字符，使模型性能受到影响。Soft-Masked BERT 的模型结构与 BFMBERT 最为相似，但是其并未利用汉字的相似性知识，且前者利用了额外的大规模数据集，从实验结果中可以看出其性能不如 BFMBERT，这证明了汉字字形和拼音特征对中文拼写检查的重要性。FASPell 利用汉字的相似度来限制解码范围，这种启发式的规则并不能参与模型整体的学习，这也限制了其性能。SpellGCN 利用更为复杂的图神经网络来建模汉字相似性知识，但与之相比 BFMBERT 更为简单和灵活，且取得了更好的效果。Chunk 使用了多种方法来构建错误纠正候选集，取得了更好的精确率，但其召回率非常低，且 BFMBERT 的 F1 值高出其 10%以上。

具体来说，相比之前性能最好的模型(SpellGCN)，BFMBERT 在错误检测上各指标分别提升了 0.48%，0.23%，0.87%和 0.6%，在错误纠正上各指标分别提升了 0.85%，0.47%，1.68%和 1.11%。

4.5　损失函数权重 λ 的影响分析

图 5 给出了损失函数权重对错误检测 F1 值和错误纠正 F1 值的影响。具体地，当 $\lambda=0.2$ 和 $\lambda=0.8$ 时，模型取得了相对较好的效果，代表在该权重下检测网络和纠正网络达到了平衡。当 λ 小于 0.2 时，BERT 并不能充分地在数据集上进行微调，这是其性能下降的主要原因。当 λ 为 0.8 时取得了次佳的效果，性能好于除 0.2 外其他权重值的原因可能是训练数据每条语句中的大部分字符是正确的，是微调过程中检测网络给出的错误概率不平衡导致的。

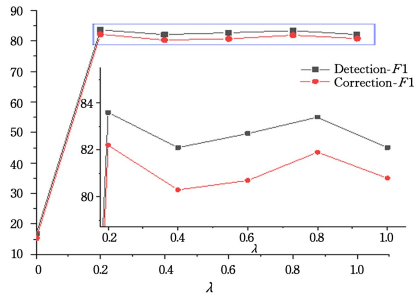


图 5　损失函数权重 λ 对模型效果的影响

Fig. 5　Influence of weight λ of loss function on model effect

4.6　拼音、字形特征融合方式研究

表 5 列出了 3.4.3 节提出的 3 种不同的拼音、字形嵌入融合方式的结果，使用两种嵌入拼接后接全连接层的模型取得了最佳的整体性能。总体来说，使用嵌入拼接的策略具有更好的性能，原因是两种嵌入代表着汉字两个特征的不同

分布，直接相加可能产生的是一个完全没有意义的向量。直接拼接比拼接后接线性层性能差的原因有两点：1）由于嵌入层维度的限制，直接拼接的方式中两种嵌入的维度大小是拼接后接线性层方式两种嵌入维度大小的一半，这限制了两种嵌入建模字形和拼音特征的能力；2）因为简体汉字作为一种表意文字，大多数形近字也是音近字，两种嵌入具有一定的相关性，拼接后使用线性层能够一定程度上模拟和融合这种相关性。

表 5 字形和拼音嵌入的融合方式比较

Table 5 Comparison of fusion methods of glyph embedding and Pinyin embedding

模型	错误检测				错误纠正			
	Acc.	*Prec.*	*Rec.*	*F1*	*Acc.*	*Prec.*	*Rec.*	*F1*
BFMBERT (cat+linear)	**84.1**	86.1	**81.3**	**83.6**	**82.9**	85.8	**78.9**	**82.2**
BFMBERT(cat)	83.9	**87.4**	79.3	83.1	82.4	**86.9**	76.2	81.2
BFMBERT(+)	82.5	84.7	79.3	81.9	81.0	84.2	76.4	80.1

4.7 消融实验

为了深入分析 BFMBERT 模型各个组件的有效性，本文在 SIGHAN2015 基准数据集上对 BFMBERT 进行了消融实验。为了避免其他因素的影响，所有消融实验的参数均相同，损失函数权重 λ 均设置为 0.2，拼音字形特征融合方式均为拼接后接线性层。具体的实验如下。

(1)BFMBERT(w/o Pinyin)：去除了拼音嵌入，即使用字形嵌入替代融合嵌入。

(2)BFMBERT(w/o Glyph)：去除了字形嵌入，即使用拼音嵌入替代融合嵌入。

(3)BFMBERT(w/o Multi-Fonts)：不使用多种字体来建模字形嵌入，仅使用宋体。

(4)BFMBERT(w/o cPretrain)：不使用 3.2 节提出的预训练策略，使用 BERT 的原始任务进行预训练。

如表 6 所列，如果去掉字形或拼音特征，模型性能的下降是非常明显的。且如前文所言，错误的字符中音近字数量多于形近字，拼音特征的重要性高于字形特征，因此去掉拼音特征的模型性能下降更为明显。使用单一字体对性能损害很严重，这说明单一字体无法准确模拟汉字的字形特征。不使用基于混淆集的预训练策略对性能的影响也很大，证明了本文提出的预训练策略使 BERT 学习到了从错字到正确字符的转换。总体来说，无论去掉哪个部分，模型的性能都会下降，但仍然明显优于 BERT，这充分地展示了 BFMBERT 每个组件和方法的有效性。

表 6 消融实验结果

Table 6 Comparison of ablation experimental results

模型	错误检测				错误纠正			
	Acc.	*Prec.*	*Rec.*	*F1*	*Acc.*	*Prec.*	*Rec.*	*F1*
BFMBERT (w/o Pinyin)	83.1	88.1	76.6	81.9	81.8	87.7	74.0	80.3
BFMBERT (w/o Glyph)	84.2	**87.3**	80.0	83.5	82.6	**86.8**	76.7	81.5
BFMBERT (w/o Multi-Fonts)	82.9	86.1	75.6	82.1	81.2	85.5	75.1	80.0
BFMBERT (w/o Pretrain)	82.6	86.1	77.8	81.8	81.4	85.7	75.3	80.2
BERT-Finetune	81.7	85.2	76.0	80.3	80.3	84.7	73.5	78.7
BFMBERT	**84.1**	86.1	**81.3**	**83.6**	**82.9**	85.8	**78.9**	**82.2**

4.8 案例研究

表 7 列出了模型处理的测试集中的两个例子。第一个例子中，“素”是错误的字符。如果不考虑汉字的相似性，可以有多个纠正候选来替换该字符，例如 BERT 输出的“换”。然而，BFMBERT 的输出是更准确的纠正，不仅是因为上下文语义中带有诉讼的隐含语境，而且“素”和“诉”具有相似的发音。第二个示例中，“官”“关”和“管”的发音都具有相似性，且“关”在语境中也有正确的语义，但 BFMBERT 的输出“管”和错误字符“官”具有相同的下部首，是更为准确的纠正，因此汉字的字形信息对于纠正拼写错误也很重要。这两个例子表明，BFMBERT 能够利用汉字的拼音和字形特征进行中文拼写检查。

表 7 案例研究

Table 7 Case study

输入文本	希望您帮我素取公平，得到他们适当的赔偿。	…青少年玩电脑越来越多，所以家长官孩子也很严…
BERT	希望您帮我换取公平，得到他们适当的赔偿。	…青少年玩电脑越来越多，所以家长关孩子也很严…
BFMBERT	希望您帮我诉取公平，得到他们适当的赔偿。	…青少年玩电脑越来越多，所以家长管孩子也很严…

结束语 本文提出了一种针对中文拼写检查任务的字词错误校对模型 BFMBERT。通过基于混淆集的掩码策略，BFMBERT 可以联合学习语义和拼写错误知识。BFMBERT 融合了汉字的拼音特征和字形特征，并在一个框架下训练检测网络和纠正网络。在基准数据集上的实验证明了模型的有效性，取得了显著的效果。在将来的工作中，我们可以考虑将模型推广到中文语法错误纠正中，另一方面将会考虑如何整合外部知识，以便模型能够处理更多真实世界的拼写错误。

参 考 文 献

[1] WU S, LIU C, LEE L. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013[C]// Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013: 35-42.

[2] YU L, LEE L, TSENG Y, et al. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check[C]// Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 126-132.

[3] TSENG Y, LEE L, CHANG L, et al. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check[C]// Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Beijing, China: Association for Computational Linguistics, 2015: 32-37.

[4] LI C, ZHANG C, ZHENG X, et al. Exploration and Exploitation: Two Ways to Improve Chinese Spelling Correction Models [C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Online: Association for Computational Linguistics, 2021: 441-446.

[5] WANG H, WANG B, DUAN J, et al. Chinese Spelling Error De-

tection Using a Fusion Lattice LSTM[J]. Transactions on Asian and Low-Resource Language Information Processing, 2021, 20(2):1-11.

[6] LIU C L, LAI M H, TIEN K W, et al. Visually and Phonologically Similar Characters in Incorrect Chinese Words: Analyses, Identification, and Applications[J]. ACM Transactions on Asian Language Information Processing, 2011, 10(2):1-39.

[7] LIU X, CHENG K, LUO Y, et al. A Hybrid Chinese Spelling Correction Using Language Model and Statistical Machine Translation with Reranking[C]// Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013: 54-58.

[8] YU J, LI Z. Chinese Spelling Error Detection and Correction Based on Language Model, Pronunciation, and Shape[C]// Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 220-223.

[9] DEVLIN J, CHANG M, LEE K, et al. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. MN, USA: Association for Computational Linguistics, 2019: 4171-4186.

[10] CUI Y, CHE W, LIU T, et al. Revisiting Pre-Trained Models for Chinese Natural Language Processing[C]// Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics, 2020: 657-668.

[11] MANGU L, BRILL E. Automatic Rule Acquisition for Spelling Correction[C]// Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1997: 187-194.

[12] CHANG T, CHEN H, YANG C. Introduction to a Proofreading Tool for Chinese Spelling Check Task of SIGHAN-8[C]// Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Beijing, China: Association for Computational Linguistics, 2015: 50-55.

[13] JIANG Y, WANG T, LIN T, et al. A Rule Based Chinese Spelling and Grammar Detection System Utility[C]// 2012 International Conference on System Science and Engineering (ICSSE). IEEE, 2012: 437-440.

[14] XIONG J, ZHANG Q, ZHANG S, et al. HANSpeller: a Unified Framework for Chinese Spelling Correction[C]// Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Beijing, China: Association for Computational Linguistics, 2015: 38-45.

[15] CHANG C. A New Approach for Automatic Chinese Spelling Correction[C]// Proceedings of Natural Language Processing Pacific Rim Symposium. Citeseer, 1995: 278-283.

[16] HUANG C, PAN H, MING Z, et al. Automatic Detecting/Correcting Errors in Chinese Text by An Approximate Word-matching Algorithm[C]// Proceedings of the 38th Annual Meeting on Association for Computational Linguistics. USA: Association for Computational Linguistics, 2000: 248-254.

[17] CHEN K, LEE H, LEE C, et al. A Study of Language Modeling for Chinese Spelling Check[C]// Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013: 79-83.

[18] XIN Y, ZHAO H, WANG Y, et al. An Improved Graph Model for Chinese Spell Checking[C]// Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 157-166.

[19] DONG S, FUNG G P C, LI B, et al. ACE: Automatic Colloquialism, Typographical and Orthographic Errors Detection for Chinese Language[C]// Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations. Osaka, Japan: The COLING 2016 Organizing Committee, 2016: 194-197.

[20] CHIU H, WU J, CHANG J S. Chinese Spell Checking Based on Noisy Channel Model[C]// Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 202-209.

[21] WANG Y, LIAO Y. Word Vector/Conditional Random Field-based Chinese Spelling Error Detection for SIGHAN-2015 Evaluation[C]// Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Beijing, China: Association for Computational Linguistics, 2015: 46-49.

[22] HUANG Q, HUANG P, ZHANG X, et al. Chinese Spelling Check System Based on Tri-gram Model[C]// Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 173-178.

[23] LIU M, JIAN P, HUANG H. Introduction to BIT Chinese Spelling Correction System at CLP 2014 Bake-off[C]// Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing. Wuhan, China: Association for Computational Linguistics, 2014: 179-185.

[24] WANG D, TAY Y, ZHONG L. Confusionset-guided Pointer Networks for Chinese Spelling Check[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 5780-5785.

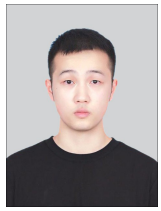
[25] TAO Y C, WU W L, HAI Z Y, el al. Text Proofreading Model with LSTM and Integrated Algorithm[J]. Journal of Chinese Computer Systems, 2020, 41(5): 967-971.

[26] YANG Z L, LI T R, LIU S J, el al. Streaming Parallel Text Proofreading Based on Spark Streaming[J]. Computer Science, 2020, 47(4): 36-41.

[27] LI C, CHEN J, CHANG J S. Chinese Spelling Check Based on Neural Machine Translation[C]// Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation. Hong Kong: Association for Computational Linguistics, 2018.

[28] MALMI E, KRAUSE S, ROTHE S, et al. Encode, Tag, Realize: High-precision Text Editing[C]// Proceedings of the 2019 Con-

ference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing(EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019: 5054-5065.
[29] LIU Y, OTT M, GOYAL N, et al. Roberta: A Robustly Optimized Bert Pretraining Approach[J]. arXiv: 1907. 11692. 2019.
[30] YANG Z, DAI Z, YANG Y, et al. Xlnet: Generalized Autoregressive Pretraining for Language Understanding[C]// Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc, 2019: 11.
[31] CLARK K, LUONG M, LE Q V, et al. Electra: Pre-training Text Encoders as Discriminators Ratherthan Generators[J]. arXiv: 2003. 10555. 2020.
[32] WANG B, CHE W, WU D, et al. Dynamic Connected Networks for Chinese Spelling Check[C]// Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Online: Association for Computational Linguistics, 2021: 2437-2446.
[33] HONG Y. FASPell: A Fast, Adaptable, Simple, Powerful Chinese Spell Checker Based On DAE-Decoder Paradigm[C]// Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019). Hong Kong, China: Association for Computational Linguistics, 2019: 160-169.
[34] CHENG X, XU W, CHEN K, et al. SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 871-881.
[35] ZHANG S, HUANG H, LIU J, et al. Spelling Error Correction with Soft-Masked BERT[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 882-890.
[36] HUANG L, LI J, JIANG W, et al. PHMOSpell: Phonological and Morphological Knowledge Guided Chinese Spelling Check [C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Online: Association for Computational Linguistics, 2021: 5958-5967.
[37] MENG Y, WU W, WANG F, et al. Glyce: Glyph-vectors for Chinese Character Representations[C]// Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019. Vancouver, BC, Canada: Curran Associates Inc, 2019: 2742-2753.
[38] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[C]// Advances in Neural Information Processing Systems. Curran Associates Inc, 2017: 5998-6008.
[39] WANG D, SONG Y, LI J, et al. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, Belgium: Association for Computational Linguistics, 2018: 2517-2527.
[40] CUI Y, CHE W, LIU T, et al. Pre-training with Whole Word Masking for Chinese Bert[J]. arXiv: 1906. 08101. 2019.
[41] LOSHCHILOV I, HUTTER F. Decoupled Weight Decay Regularization[J]. arXiv: 1711. 05101. 2017.
[42] XIE H H, LI A L, LI Y B, et al. CPLM-CSC: Character-based Pre-trained Language Model for Chinese Spelling Checking and Correction[J]. Journal of Chinese Information Processing, 2021, 35(5): 38-45.
[43] BAO Z, LI C, WANG R. Chunk-based Chinese Spelling Check with Global Optimization[C]// Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics, 2020: 2031-2040.
[44] LIU S, YANG T, YUE T, et al. PLOME: Pre-training with Misspelled Knowledge for Chinese Spelling Correction[C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Online: Association for Computational Linguistics, 2021: 2991-3000.



LIU Zhe, born in 1998, postgraduate, is a member of China Computer Federation. His main research interests include Chinese spelling check, Chinese grammatical error correction and natural language processing.

LI Tianrui, born in 1969, Ph.D, professor, Ph.D supervisor, is a distinguished member of China Computer Federation. His main research interests include big data intelligence, rough sets and granular computing.

(责任编辑:喻藜)