基于中文语义−音韵信息的语音识别文本校对模型

仲美玉[1]，吴培良[1,2]，窦燕[1,3]，刘毅[1]，孔令富[1,2]
（1. 燕山大学信息科学与工程学院，河北 秦皇岛 066004；
2. 河北省计算机虚拟技术与系统集成重点实验室，河北 秦皇岛 066004；
3. 河北省软件工程重点实验室，河北 秦皇岛 066004）

摘 要：为了研究拼音对检测和纠正语音识别文本错误的影响，提出了一种基于中文语义−音韵信息的文本校对模型。定义了 5 种拼音编码方法构建字符−音韵嵌入向量，以此作为基于 GRU 的 Seq2Seq 模型的输入，并应用注意力机制提取语句的语义−音韵信息来校对语音识别文本错误。针对标注语料不足的问题，提出了一种基于拼音声韵置换的数据增强方法。在 AISHELL-3 公开数据集的实验结果表明，拼音携带的音韵信息有利于校对语音识别文本错误，所提方法可提升模型的检错性能。

关键词：文本校对；语音识别；拼音；注意力机制
中图分类号：TP391
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022222

Chinese semantic and phonological information-based text proofreading model for speech recognition

ZHONG Meiyu[1], WU Peiliang[1,2], DOU Yan[1,3], LIU Yi[1], KONG Lingfu[1,2]
1. School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
2. The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao 066004, China
3. The Key Laboratory of Software Engineering of Hebei Province, Qinhuangdao 066004, China

Abstract: To study the influence of Chinese Pinyin on detecting and correcting text errors in speech recognition, a text proofreading model based on Chinese semantic and phonological information was proposed. Five Pinyin coding methods were designed to construct the character-Pinyin embedding vector that was employed as the input of the Seq2Seq model based on gated recurrent unit. At the same time, the attention mechanism was adopted to extract the Chinese semantic and phonological information of sentences to correct speech recognition errors. Aiming at the problem of insufficient labeled corpus, a data augmentation method was introduced, which could automatically obtain annotated corpora by exchanging the initials or finals of Chinese Pinyin. The experimental results on AISHELL-3's public data show that phonological information is conducive to the text proofreading model to detect and correct text errors after speech recognition, and the proposed data augmentation method can improve the error detection performance of the model.

Keywords: text proofreading, speech recognition, Pinyin, attention mechanism

---

收稿日期：2022–07–28；**修回日期：**2022–10–24
通信作者：吴培良，peiliangwu@ysu.edu.cn
基金项目：国家重点研发计划基金资助项目（No.2018YFB1308300）；国家自然科学基金资助项目（No.62276028, No.U20A20167）；北京市自然科学基金资助项目（No.4202026）；河北省自然科学基金资助项目（No.F202103079）；河北省创新能力提升计划基金资助项目（No.22567626H）；河北省软件工程重点实验室基金资助项目（No.22567637H）
Foundation Items: The National Key Research and Development Program of China (No.2018YFB1308300), The National Natural Science Foundation of China (No.62276028, No.U20A20167), Beijing Natural Science Foundation (No.4202026), The Natural Science Foundation of Hebei Province (No.F202103079), The Innovation Capability Improvement Plan Project of Hebei Province (No.22567626H), The Project of the Key Laboratory of Software Engineering of Hebei Province (No.22567637H)

0 引言

近年来，自动语音识别（ASR, automatic speech recognition）技术被广泛地应用于人机交互系统。受用户发音不清晰、环境噪声等因素的影响，实际应用场景下的语音识别准确率仍然不高[1]。中文存在大量发音相近但意义完全不同的汉语字符，语言自身的复杂性进一步导致了语音识别错误的产生[2]。从语音识别文本长度变化的角度分析，ASR 系统产生的文本错误包括多字错误、少字错误和替换错误 3 种类型。从语音识别文本发音变化的角度分析，语音识别文本中存在大量谐音错误，如图 1 所示，“镜”被误识为“睛”。除此之外，语音识别文本中还存在混淆音错误，如图 1 中“牛郎”被误识为“流浪”。ASR 模块通常位于人机语音交互系统前端，语音识别错误文本反馈至交互界面会增加用户理解语义的难度，也会直接影响意图识别、命名实体识别等下游任务的处理[3]。语音识别后的文本校对能有效避免识别错误在 ASR 系统下游任务的累积，是改进 ASR 系统性能的重要方法[4]。

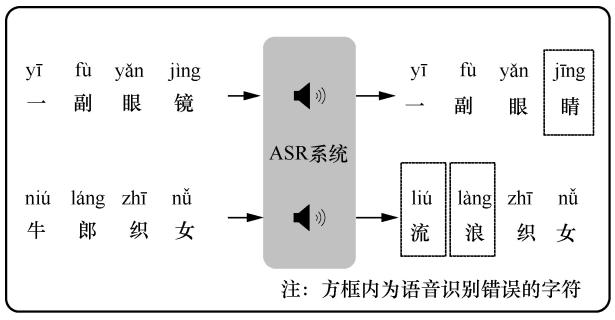

图 1 语音识别错误示例

替换错误在语音识别错误中占有较大比重[5]，故本文侧重于检测和纠正语音识别文本中的替换错误。中文文本校对方法主要分为 3 种，分别是基于规则的校对方法、基于统计的校对方法和基于深度学习的校对方法。相较基于规则的校对方法和基于统计的校对方法，基于深度学习的校对方法能够捕获更深层次的语义信息，有利于提升文本校对效果[6]。针对现有基于深度学习的模型只考虑使用文本的语义信息纠正错误字符的问题，Chen 等[7]构建了融合语义信息和音韵信息的预训练语言模型来实现语音识别文本校对，该方法首先使用微调的预训练语言模型定位语句中错误字符的位置，采用掩码字符掩盖错误字符，利用模型提取的语义信息计算纠错候选字符的概率；然后，采用 DIMSIM[8]计算错误字符与各候选字符的拼音距离；最后，综合考虑候选字符的概率及其与错误字符的拼音距离来完成文本纠错，实验证明了利用拼音信息能够有效地加强模型纠正语音识别文本中谐音错误字符的能力。Duan 等[9]使用一维卷积神经网络（1D-CNN, one-dimensional convolutional neural network）构建的序列到序列（Seq2Seq, sequence to sequence）模型来校对文本，该方法采用字节对编码方法生成拼音的嵌入向量（以下统称音韵嵌入向量）并将其作为模型的输入，以便模型提取并利用语句的音韵信息纠正文本错误，实验验证了字粒度切分语句的方式和带声调的拼音有助于提高语音识别文本纠错效果。综上所述，拼音携带的音韵信息对纠正 ASR 系统识别错误具有重要意义。由于拼音中的字母不可随意调换顺序，例如，图 1 中“郎”字的拼音是“láng”，调换其拼音中任意 2 个字母的位置后（如“lnág”、“lágn”），便不再是“郎”字的读音，故中文字符的拼音本质上是一种序列。然而，上述工作在生成音韵嵌入向量时没有保留拼音的时序信息，也没有对生成音韵嵌入向量的拼音编码方法及其对检测和纠正语音识别文本错误的影响做进一步研究。此外，基于深度学习的文本校对模型往往需要通过大量的标注语料来增强其对文本语义和文本结构信息的学习能力，从而提升模型的检错和纠错性能。但实际应用中的 ASR 系统通常面向垂直领域，可获取的标注语料十分有限。虽然可以使用其他语料库来扩充数据集，但该方式不能促使模型学习到更多与特定对话场景相关的文本语义和文本结构信息。

为了解决上述问题，本文提出了 5 种拼音编码方法来生成中文字符的含拼音时序信息的音韵嵌入向量，分别将各个拼音编码方法与带有注意力机制的编码器−解码器架构相结合来建立基于中文语义−音韵信息（CSPI, Chinese semantic and phonological information）的文本校对模型；从汉语拼音组成成分的角度分析了语音识别文本错误的特点，并据此提出了一种基于拼音声韵置换（RPIF, replacement of Pinyin’s initials or finals）的数据增强方法，该方法可利用有限的语料来生成大量的纠错数据，以便利用数据驱动的方法构建面向垂直领域的文本校对模型。本文的主要贡献可以总结为以下 4 点。

1) 提出了 5 种拼音编码方法来生成中文字符的音韵嵌入向量。所提方法采用不同的处理时序数据的神经网络来编码拼音序列，从而以多种方式生成含有拼音时序信息的音韵嵌入向量，便于研究不同拼音编码方法对语音识别文本校对任务的影响。

2) 构建了基于 CSPI 的语音识别文本校对模型。该模型由上述拼音编码方法分别与带有注意力机制的编码器–解码器架构组合而成，能充分地提取并利用中文语句的语义和音韵信息校对语音识别文本错误。

3) 提出了一种基于 RPIF 的数据增强方法。该方法能够有效模拟用户因发音不清晰、口误等造成的语音识别错误，解决了因标注语料不足而难以面向特定对话场景构建基于深度学习的文本校对模型的问题。

4) 在多人普通话语音数据集 AISHELL-3 上开展了相关实验，验证了拼音携带的音韵信息有利于文本校对模型检测和纠正语音识别文本错误，归纳了不同的拼音编码方法对检测和纠正语音识别文本错误的影响。

1　相关工作

语音识别后的文本校对是提升 ASR 系统性能的重要方法。文献[1]综述了 ASR 系统识别错误的产生原因和处理方法。早期的研究主要是对语音识别错误检测方法的研究，对语音识别错误纠正方法的研究则相对较少。中文文本校对方法可分为 3 种：基于规则的校对方法、基于统计的校对方法和基于深度学习的校对方法。文献[10-11]均通过观察文本错误出现的规律并制定相应的规则来处理文本错误。此类基于规则的校对方法仅对特定的错误类型有效，其文本校对效果也严重依赖于规则制定的好坏[12-13]。现有 ASR 系统在实际对话场景中产生的识别错误具有较强的复杂性，无法使用简单的规则覆盖所有可能出现的错误。N-gram 是文本校对任务中最常用的统计语言模型[14]。文献[15]使用 N-gram 语言模型和潜在语义分析方法相结合的方式来校对文本错误。文献[16]建立了基于 2-gram 和 3-gram 的文本校对方法，并采用了平滑技术来解决数据稀疏的问题。文献[17]结合使用语言模型和统计机器翻译方法生成错误字符的候选集，采用支持向量机对候选集排序的方式实现中文语句的自动校对。然而，基于统计的校对方法在使用混淆集纠正文本错误时，没有充分利用句子的上下文语义关系，容易出现邻近词正确，但整个句子不符合逻辑的情况。因此，上述基于规则和基于统计的文本校对方法均难以有效地处理 ASR 系统实际应用过程中出现的语音识别错误。近年来，越来越多的研究将深度学习技术运用到中文文本处理任务中，基于深度神经网络的文本校对方法也不断被提出[18-21]。文献[22]将检测文本错误字符的问题视为序列标注问题，利用双向长短期记忆（LSTM, long-short term memory）网络构建了拼写文本检错模型。文献[23]构建了基于双向 LSTM 的 Seq2Seq 模型来检测和纠正文本中的错误字符。文献[24]构建了基于 1D-CNN 的 Seq2Seq 模型来实现文本校对。基于深度学习的校对方法能利用深度神经网络模型捕获更丰富的文本语义和文本结构信息来校对文本错误，通常能取得比基于规则和基于统计的校对方法更好的检错和纠错效果。

语音识别文本校对和拼写文本校对的研究目标一致，本质上都是检测和纠正文本中的错误字符。中文拼写错误主要来源于人们错误使用了某个字符的谐音或形似字符[25]。近年来，一些研究工作尝试利用文本的拼音和字形信息来提升基于深度学习的拼写文本校对模型的性能。Wang 等[26]构建了基于 Lattice LSTM 和 CRF 的拼写错误检测模型，该模型融合字符、词语和拼音 3 种信息进行错误检测，验证了拼音信息有利于检测拼写错误。Liu 等[27]提出了使用单向门控循环单元（Uni-GRU, unidirectional gated recurrent unit）编码字符的无声调拼音和笔画来获取更有意义的字符表示，并以此作为预训练语言模型的输入。实验结果表明，融合拼音和笔画信息的预训练模型在拼写文本校对任务中表现出了十分优异的性能。与之类似，文献[28-32]也提出了多种基于深度学习的拼写文本校对方法，部分研究工作以不同方式利用字符的音韵信息来提升模型性能。表 1 列举了多项研究在 SIGHAN2015 拼写纠错数据集[33]上的评估结果。从表 1 可以看出，基于深度学习的拼写校对模型通常比基于统计的拼写校对模型有更好的检错和纠错效果，字符的音韵信息对提升拼写校对模型的检错和纠错性能有积极影响。相较于拼写文本错误，语音识别文本错误不仅包含谐音类型的错误字符，还包含较多因用户发音不清晰、环境嘈杂等因素导致的混淆音类型的错误字符。然而，现有面向语音识别文本校对任务的相关工作没有深入地研究拼音所蕴含的音韵信息对检测和纠正语音识别文本错误的影响。考虑到

表 1 多项研究在 SIGHAN2015 拼写纠错数据集上的评估结果

模型简称	音韵信息	基于深度学习的模型	检错 F1 值	纠错 F1 值
NTOU[33]	×	×	42.01%	36.64%
NCTU-NTUT[33]	×	×	45.79%	37.55%
Fusion Lattice LSTM-CRF[26]	√	√	49.10%	—
Confusionset[23]	×	√	69.80%	64.90%
FASPell[28]	×	√	63.50%	62.60%
Soft-Masked BERT[29]	×	√	73.50%	66.40%
SpellGCN[30]	√	√	77.70%	75.90%
SpellBERT[31]	√	√	80.00%	78.50%
MLM-phonetics[32]	√	√	80.20%	77.50%

汉语拼音是一种序列且带声调的拼音能完整地保留字符音韵信息，本文参考文献[27]提出了一种新的基于 Uni-GRU 的拼音编码方法，同时又基于 1D-CNN、双向门控循环单元（Bi-GRU, bidirectional gated recurrent unit）等处理时序数据的网络提出了 4 种拼音编码方法来编码带声调的拼音序列，以生成保留完整音韵信息的嵌入向量。将各个拼音编码方法分别与带有注意力机制的编码器–解码器架构相结合来构建基于 CSPI 的文本校对模型，以明确有利于检测和纠正语音识别文本错误的拼音编码方法。

由于标注数据有限，许多先进的深度学习模型难以被有效地应用于文本校对任务。为了满足通过大量标注数据提升模型校对性能的需求，Wang 等[22]利用基于光学字符识别和自动语音识别的方法模拟拼写错误，实现了面向拼写纠错任务的数据增强方法。Liu 等[27]和 Cheng 等[30]通过上述数据增强方法生成的语料构建了大规模预训练语言模型，该模型在拼写纠错任务中取得了非常优异的成绩。然而，ASR 系统识别错误比拼写错误更复杂，主要原因是 ASR 系统在用户发音不清晰或环境嘈杂的情况下获取了含较多噪声的声音信号，ASR 系统的语言模型因受噪声干扰无法将声音信号解码为正确的文本序列。值得注意的是，Wang 等[22]提出的数据增强方法根据拼写错误的特点摒弃了语音识别过程中真实产生的混淆音类别的错误文本。其他面向拼写纠错任务的数据集也存在包含较少混淆音类别的错误文本的问题。这意味着在拼写纠错数据集上表现出色的文本校对模型不一定在语音识别后的文本校对任务中具备同等优秀的纠错能力。因此，本文从汉语拼音组成成分的角度分析 ASR 系统识别错误的特点，并据此提出一种基于 RPIF 的数据增强方法，以便将先进的深度学习模型应用于语音识别后的文本校对任务中，进而辅助 ASR 系统提升其识别准确性。

2 基于 CSPI 的文本校对模型

基于 CSPI 的文本校对模型受启发于神经机器翻译模型[34-35]，使用带有注意力机制的编码器–解码器架构[36]来实现错误文本到正确文本的转换，模型的总体结构如图 2 所示。首先，使用常见的处理时序型数据的神经网络编码中文字符的拼音序列，生成含时序信息的音韵嵌入向量。然后，分别融合错误文本中各个字符的音韵嵌入向量和字符嵌入向量，以此作为编码器的输入。接着，编码器编码错误文本，输出错误文本的语义–音韵向量，该语义–音韵向量则携带了错误文本全部的语义–音韵信息。最后，解码器以语义–音韵向量和解码起始符为输入，先采用注意力机制捕获当前解码字符与错误文本的上下文语义–音韵关系，再利用该语义–音韵关系输出预测字符，进而逐步解码预测的正确文本。

接下来，先从数学角度定义模型校对语音识别错误文本的过程，再从拼音编码、编码器、解码器和优化目标 4 个方面详细介绍基于 CSPI 的文本校对模型。

2.1 问题定义

假设错误文本为源（source）文本序列 $\boldsymbol{s}=\{s_1,\cdots,s_i,\cdots,s_n\}$，文本校对模型输出的语句是目标（target）文本序列 $\boldsymbol{g}=\{g_1,\cdots,g_t,\cdots,g_m\}$。从概率角度分析，文本校对的过程相当于给定 $\boldsymbol{s}$，寻找 $\boldsymbol{g}$ 来最大化条件概率 $p(\boldsymbol{g}|\boldsymbol{s})$。因此，文本校对的目标是建立一个参数化模型，使用平行语料库来训练该模

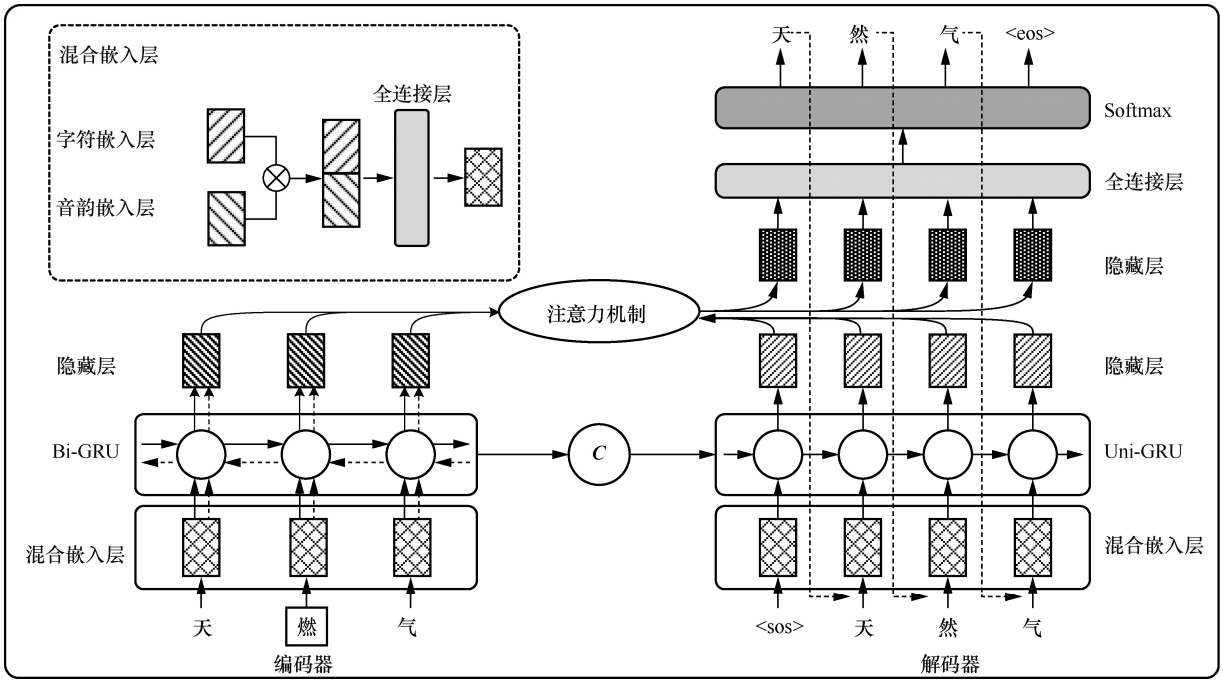

图 2　基于 CSPI 的文本校对模型的总体结构

型，以最大化各个 source-target 语句对的条件概率。当模型学习到这个条件概率分布后，给定一个错误文本，模型便可以输出一个条件概率最大的句子作为预测的正确文本。为了利用句子的音韵信息来加强模型校对语音识别错误文本的能力，本文提出了 5 种拼音编码方法来构建基于 CSPI 的文本校对模型。假设 $\boldsymbol{s}$ 对应的拼音序列为 $\boldsymbol{s}_{\mathrm{p}}=\{s_1^{\mathrm{p}},\cdots,s_i^{\mathrm{p}},\cdots,s_n^{\mathrm{p}}\}$，则 $p(\boldsymbol{g}|\boldsymbol{s})$ 的求解过程转化为 $p(\boldsymbol{g}|\boldsymbol{s},\boldsymbol{s}_{\mathrm{p}})$。

2.2　拼音编码

拼音是由小写拉丁字母构成的汉字发音标记，一般包含声母、韵母和声调 3 个部分，如图 3 所示。为了便于计算机识别，将图 3 中 4 种声调依次映射到数字{1,2,3,4}，则 4 个汉字的拼音可表示为‘fei1, yan2, zou3, bi4’。除了图 3 所示的 4 种声调外，中文还存在轻声这一特殊的声调。‘轻声’字符的拼音不标注声调，仅由小写拉丁字母构成，例如，‘云彩’中的‘彩’为轻声，其拼音为‘cai’。

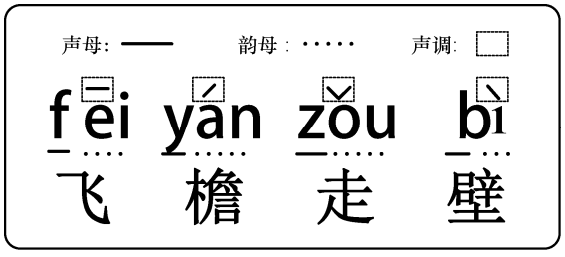

图 3　汉语拼音示例

为了建模字符间的音韵关系，本文将字符拼音视为由小写字母和声调组成的序列，使用不同的处理时序数据的神经网络（Uni-GRU、Bi-GRU 和 1D-CNN）编码拼音序列，由此获取含时序信息的音韵嵌入向量，以使音似字符间有相似的音韵表示。在之前的研究工作中，Duan 等[9]验证了字粒度切分方式和带声调的拼音序列有利于语音识别文本纠错，因此，本文采用字粒度切分方式划分语句，

使用 PyPinyin 工具包获取各个字符的带有声调的拼音序列。本文将拼音序列的长度固定为 8，当拼音序列的实际长度未达到 8 时使用数字‘0’填充。根据编码拼音序列的网络类型的不同，将本文提出的 5 种拼音编码方法分别命名为 $\mathrm{P_C}$、$\mathrm{P_U}$、$\mathrm{P_B}$、$\mathrm{P_{CU}}$ 和 $\mathrm{P_{CB}}$。图 4 以‘中’的拼音‘zhong1’为例，示意了上述 5 种拼音编码方法。由图 4 可知，$\mathrm{P_C}$、$\mathrm{P_U}$ 和 $\mathrm{P_B}$ 使用一种类型的神经网络编码拼音序列，本文将其统称为单网络拼音编码方法。$\mathrm{P_{CU}}$ 和 $\mathrm{P_{CB}}$ 使用 2 种不同类型的神经网络编码拼音序列，以获取更加全面的音韵信息，本文将其统称为混合网络拼音编码方法。以下是对 5 种拼音编码方法的定义。

定义 1　$\mathrm{P_C}$ 拼音编码。对于任意一个中文字符 c 的拼音序列 c_{p}，使用单层 1D-CNN 编码 c_{p}，生成字符 c 的 $\mathrm{P_C}$ 音韵嵌入向量，即

$$\boldsymbol{p}_{\mathrm{C}}(c_{\mathrm{p}})=\mathrm{Maxpool}(\varphi_{\mathrm{CNN}}(E(c_{\mathrm{p}}))) \tag{1}$$

其中，φ_{CNN} 是单层 1D-CNN 的函数表示，Maxpool 指最大池化层，E 指字符嵌入层。

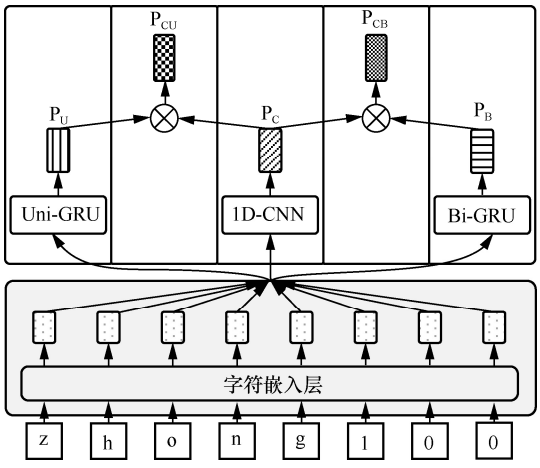

图 4　拼音编码

定义 2　P_U 拼音编码。对于任意一个中文字符 c 的拼音序列 c_p，使用单层 Uni-GRU 网络编码 c_p，生成字符 c 的 P_U 音韵嵌入向量，即

$$\boldsymbol{p}_\text{U}(c_\text{p})=\varphi_{\text{Uni-GRU}}(E(c_\text{p})) \tag{2}$$

其中，$\varphi_{\text{Uni-GRU}}$ 是单层 Uni-GRU 网络的函数表示。

定义 3　P_B 拼音编码。对于任意一个中文字符 c 的拼音序列 c_p，使用单层 Bi-GRU 网络编码 c_p，生成字符 c 的 P_B 音韵嵌入向量，即

$$\boldsymbol{p}_\text{B}(c_\text{p})=\varphi_{\text{Bi-GRU}}(E(c_\text{p})) \tag{3}$$

其中，$\varphi_{\text{Bi-GRU}}$ 是单层 Bi-GRU 网络的函数表示。

定义 4　P_CU 拼音编码。对于任意一个中文字符 c 的拼音序列 c_p，融合 $\boldsymbol{p}_\text{C}$ 和 $\boldsymbol{p}_\text{U}$ 编码 c_p 所得结果，生成字符 c 的 P_CU 音韵嵌入向量，即

$$\boldsymbol{p}_\text{CU}(c_\text{p})=f([\boldsymbol{p}_\text{C}(c_\text{p});\boldsymbol{p}_\text{U}(c_\text{p})]) \tag{4}$$

其中，f 表示全连接（FC, fully connected）层，[·] 表示合并操作。

定义 5　P_CB 拼音编码。对于任意一个中文字符 c 的拼音序列 c_p，融合 $\boldsymbol{p}_\text{C}$ 和 $\boldsymbol{p}_\text{B}$ 编码 c_p 所得结果，生成字符 c 的 P_CB 音韵嵌入向量，即

$$\boldsymbol{p}_\text{CB}(c_\text{p})=f([\boldsymbol{p}_\text{C}(c_\text{p});\boldsymbol{p}_\text{B}(c_\text{p})]) \tag{5}$$

由图 4 可得，c_p 先通过字符嵌入层获取其字母或声调的嵌入向量，而后任选一种拼音编码方法来生成字符 c 的音韵嵌入向量 $\boldsymbol{e}_c^\text{p}$，即

$$\boldsymbol{e}_c^\text{p}\in\{\boldsymbol{p}_\text{C}(c_\text{p}),\boldsymbol{p}_\text{U}(c_\text{p}),\boldsymbol{p}_\text{B}(c_\text{p}),\boldsymbol{p}_\text{CU}(c_\text{p}),\boldsymbol{p}_\text{CB}(c_\text{p})\} \tag{6}$$

2.3　编码器

编码器由混合嵌入（FE, fusion embedding）层和单层 Bi-GRU 网络构成，负责输出源文本序列 $\boldsymbol{s}$ 在各个时间步的隐藏（Hidden）层及其语义−音韵向量 $\boldsymbol{C}$，其结构如图 2 所示。构建混合嵌入层旨在建立中文句子及其拼音序列间的关系。选用 Bi-GRU 是希望编码器能通过该网络充分提取 $\boldsymbol{s}$ 的上下文语义−音韵信息。

首先，源文本序列 $\boldsymbol{s}$ 及其拼音序列 $\boldsymbol{s}_\text{p}$ 经混合嵌入层后得到字符−音韵嵌入向量 $\boldsymbol{e}_s^\text{cp}$

$$\boldsymbol{e}_s^\text{cp}=E_\text{cp}(\boldsymbol{s},\boldsymbol{s}_\text{p})=[\boldsymbol{e}_{s_1}^\text{cp},\cdots,\boldsymbol{e}_{s_i}^\text{cp},\cdots,\boldsymbol{e}_{s_n}^\text{cp}] \tag{7}$$

其中，E_cp 是混合嵌入层的函数表示，$\boldsymbol{e}_{s_i}^\text{cp}$ 是 $\boldsymbol{s}$ 中任意一个中文字符 s_i 的字符−音韵嵌入向量，如式(8)所示。

$$\boldsymbol{e}_{s_i}^\text{cp}=f([\boldsymbol{e}_{s_i}^\text{c};\boldsymbol{e}_{s_i}^\text{p}]) \tag{8}$$

其中，$\boldsymbol{e}_{s_i}^\text{c}=E(s_i)$ 是 s_i 的字符嵌入向量，$\boldsymbol{e}_{s_i}^\text{p}\in\{\boldsymbol{p}_\text{C}(s_i^\text{p}),\boldsymbol{p}_\text{U}(s_i^\text{p}),\boldsymbol{p}_\text{B}(s_i^\text{p}),\boldsymbol{p}_\text{CU}(s_i^\text{p}),\boldsymbol{p}_\text{CB}(s_i^\text{p})\}$ 是 s_i 的音韵嵌入向量。

然后，将字符−音韵嵌入向量 $\boldsymbol{e}_s^\text{cp}$ 输入 Bi-GRU 层。具体地，Bi-GRU 层的前向网络和反向网络分别以正序 $\boldsymbol{e}_s^\text{cp}$（从 $\boldsymbol{e}_{s_1}^\text{cp}$ 到 $\boldsymbol{e}_{s_n}^\text{cp}$）和逆序 $\boldsymbol{e}_s^\text{cp}$（从 $\boldsymbol{e}_{s_n}^\text{cp}$ 到 $\boldsymbol{e}_{s_1}^\text{cp}$）为输入，输出前向隐藏状态 $\vec{\boldsymbol{h}}_s$ 和反向隐藏状态 $\overleftarrow{\boldsymbol{h}}_s$，如式(9)和式(10)所示。

$$\vec{\boldsymbol{h}}_s=[\vec{\boldsymbol{h}}_{s_1},\cdots,\vec{\boldsymbol{h}}_{s_i},\cdots,\vec{\boldsymbol{h}}_{s_n}] \tag{9}$$

$$\overleftarrow{\boldsymbol{h}}_s=[\overleftarrow{\boldsymbol{h}}_{s_1},\cdots,\overleftarrow{\boldsymbol{h}}_{s_i},\cdots,\overleftarrow{\boldsymbol{h}}_{s_n}] \tag{10}$$

其中，$\vec{\boldsymbol{h}}_{s_i}$ 和 $\overleftarrow{\boldsymbol{h}}_{s_i}$ 分别是 Bi-GRU 层在 i 时刻输出的前向和反向隐藏状态。合并 Bi-GRU 在 i 时刻输出的前向和反向隐藏状态可得编码器在每个时间步输出的隐藏状态 $\boldsymbol{h}_{s_i}$，如式(11)所示。

$$\boldsymbol{h}_{s_i}=\sigma(f([\vec{\boldsymbol{h}}_{s_i};\overleftarrow{\boldsymbol{h}}_{s_i}])) \tag{11}$$

其中，σ 表示激活函数 tanh。则编码器在各个时间步输出的隐藏状态 $\boldsymbol{h}_s$ 可表示为

$$\boldsymbol{h}_s=[\boldsymbol{h}_{s_1},\cdots,\boldsymbol{h}_{s_i},\cdots,\boldsymbol{h}_{s_n}] \tag{12}$$

其中，$\boldsymbol{h}_{s_i}$ 包含整个源文本序列的语义−音韵信息且主要包含源指令第 i 个字符及其周边字符的语义−音韵信息。

根据文献[35]，本文使用编码器在最后一个时间步上的隐藏状态作为源文本序列 $\boldsymbol{s}$ 的语义−音韵向量 $\boldsymbol{C}$，即

$$\boldsymbol{C}=\boldsymbol{h}_{s_n} \tag{13}$$

2.4　解码器

解码器由混合嵌入层和单层的 Uni-GRU 网络构成，使用源文本序列的语义−音韵向量 $\boldsymbol{C}$ 初始化 Uni-GRU 层的隐藏状态，采用注意力机制输出预测的文本序列，其结构如图 2 所示。

在模型训练阶段，本文采用 teach-forcing 训练策略。将输入解码器的文本序列及其拼音序列分别表示为 $\boldsymbol{g}=\{g_1,\cdots,g_t,\cdots\}$ 和 $\boldsymbol{g}_{\mathrm{p}}=\{g_1^{\mathrm{p}},\cdots,g_t^{\mathrm{p}},\cdots\}$，此时，$g_1=g_1^{\mathrm{p}}=<\mathrm{sos}>$，$g_t=g_{t-1}$，$g_t^{\mathrm{p}}=g_{t-1}^{\mathrm{p}}$。$\boldsymbol{g}$ 和 $\boldsymbol{g}_{\mathrm{p}}$ 经混合嵌入层可生成字符−音韵嵌入向量 $\boldsymbol{e}_g^{\mathrm{cp}}$，结合式(7)和式(8)可得

$$\boldsymbol{e}_g^{\mathrm{cp}}=E_{\mathrm{cp}}(\boldsymbol{g},\boldsymbol{g}_{\mathrm{p}})==[\boldsymbol{e}_{g_1}^{\mathrm{cp}},\cdots,\boldsymbol{e}_{g_t}^{\mathrm{cp}},\cdots] \tag{14}$$

Uni-GRU 层以字符−音韵嵌入向量 $\boldsymbol{e}_g^{\mathrm{cp}}$ 为输入，输出隐藏状态 $\boldsymbol{h}_g$，结合式(9)可得

$$\boldsymbol{h}_g=[\boldsymbol{h}_{g_1},\cdots,\boldsymbol{h}_{g_t},\cdots]=[\vec{\boldsymbol{h}}_{g_1},\cdots,\vec{\boldsymbol{h}}_{g_t},\cdots] \tag{15}$$

其中，$\boldsymbol{h}_{g_t}$ 是 Uni-GRU 层在 t 时刻输出的隐藏状态。

本文采用注意力机制[37]使解码器在动态解码过程中，给予源文本序列中与目标字符相关性较高的字符以较大权重，以便模型能准确输出目标文本序列。以编码器和解码器在各个时间步输出的隐藏状态 $\boldsymbol{h}_s$ 和 $\boldsymbol{h}_g$ 作为注意力机制输入，将注意力机制在 t 时刻输出的隐藏状态记为 $\tilde{\boldsymbol{h}}_{g_t}$，其计算方法如式(16)所示。

$$\tilde{\boldsymbol{h}}_{g_t}=\sigma(f([\boldsymbol{h}_{g_t};\boldsymbol{c}_t])) \tag{16}$$

其中，$\boldsymbol{c}_t$ 是编码器输出的各个隐藏状态在 t 时刻的加权平均和，可表示为

$$\boldsymbol{c}_t=\sum_{i=1}^{n}a_{ti}\boldsymbol{h}_{s_i} \tag{17}$$

权重 a_{ti} 的计算式为

$$a_{ti}=\frac{\exp(\mathrm{score}(\boldsymbol{h}_{g_t},\boldsymbol{h}_{s_i}))}{\sum_{i=1}^{n}\exp(\mathrm{score}(\boldsymbol{h}_{g_t},\boldsymbol{h}_{s_i}))} \tag{18}$$

其中，$\mathrm{score}(\boldsymbol{h}_{g_t},\boldsymbol{h}_{s_i})$ 参考 general 操作[37]，如式(19)所示。

$$\mathrm{score}(\boldsymbol{h}_{g_t},\boldsymbol{h}_{s_i})=\boldsymbol{h}_{g_t}{}^{\mathrm{T}}\boldsymbol{W}_a\boldsymbol{h}_{s_i} \tag{19}$$

由图 2 可知，解码器的输出层分别对目标文本序列中的每个字符及其拼音序列做出预测。模型对第 t 个字符及其拼音序列的预测概率可分别表示为

$$p_{\mathrm{c}}(g_t=x|\boldsymbol{C},g_t,g_t^{\mathrm{p}})=\mathrm{softmax}(\boldsymbol{W}_{\mathrm{c}}\tilde{\boldsymbol{h}}_{g_t}+\boldsymbol{b}_{\mathrm{c}})[x] \tag{20}$$

$$p_{\mathrm{p}}(g_t^{\mathrm{p}_j}=y|\boldsymbol{C},g_t,g_t^{\mathrm{p}})=\mathrm{softmax}(\boldsymbol{W}_{\mathrm{p}}\tilde{\boldsymbol{h}}_{g_t}+\boldsymbol{b}_{\mathrm{p}})[y] \tag{21}$$

其中，$p_{\mathrm{c}}(g_t=x|\boldsymbol{C},g_t,g_t^{\mathrm{p}})$ 是模型将目标字符 g_t 预测为词汇表第 x 个字符的条件概率，$p_{\mathrm{p}}(g_t^{\mathrm{p}_j}=y|\boldsymbol{C},g_t,g_t^{\mathrm{p}})$ 是模型将目标字符拼音序列 g_t^{p} 的第 j 字母预测为拼音词汇表第 y 个字母的条件概率，$\boldsymbol{W}_{\mathrm{c}}$、$\boldsymbol{b}_{\mathrm{c}}$、$\boldsymbol{W}_{\mathrm{p}}$ 和 $\boldsymbol{b}_{\mathrm{p}}$ 均为全连接网络的参数。

在模型评估阶段，解码器仅以解码起始符<sos>为输入，此后的每个时间步输出一个预测的目标字符，并以该字符及其拼音序列作为解码器在下一时刻的输入，如此循环迭代，直至输出解码终止符<eos>后停止解码。每个时间步输出的预测目标字符均为词汇表中概率最高的字符。根据文献[30]，本文以预测目标字符是否与真实目标字符相匹配来实现错误检测。

2.5　优化目标

一般来说，文本纠错模型在训练阶段只设置字符优化目标。本文提出的基于 CSPI 的文本校对模型同时学习了句子的语义信息和音韵信息，因此设置了字符−拼音优化目标，如式(22)所示。

$$\mathcal{L}_{\mathrm{cp}}=\mathcal{L}_{\mathrm{c}}+\mathcal{L}_{\mathrm{p}} \tag{22}$$

其中，$\mathcal{L}_{\mathrm{c}}$ 和 $\mathcal{L}_{\mathrm{p}}$ 分别是字符优化目标和拼音优化目标，可表示为

$$\mathcal{L}_{\mathrm{c}}=-\sum_{t=1}^{m}\log p_{\mathrm{c}}(g_t=c_t|\boldsymbol{s},\boldsymbol{s}_{\mathrm{p}}) \tag{23}$$

$$\mathcal{L}_{\mathrm{p}}=-\sum_{t=1}^{m}\sum_{j=1}^{8}\log p_{\mathrm{p}}(g_t^{\mathrm{p}_j}=y_t^j|\boldsymbol{s},\boldsymbol{s}_{\mathrm{p}}) \tag{24}$$

其中，c_t 是 g_t 的正确预测字符，y_t^j 是拼音序列 g_t^{p} 第 j 个字母的正确预测字母。

3　面向中文 ASR 系统的纠错数据增强方法

本节首先根据 2.2 节所述汉语拼音的组成部分来分析语音识别错误的特点，然后根据该特点提出一种基于 RPIF 的纠错数据增强方法。

3.1 语音识别文本错误分析

表 2 列举了 Kaldi 语音识别工具包使用过程中出现的错误示例[22]。接下来，根据拼音的组成部分，即声母、韵母和声调，分析表 2 所列语音识别错误示例。示例 1 中，“幸”被误识为“行”，二者的声母和韵母均相同，声调“4”被误识为声调“2”。示例 2 中，语音识别错误字符与正确字符有着完全不同的发音，但仔细分析可以发现，“围”和“没”有相同的韵母“ei”和声调“2”，语音识别错误来源于声母“m”被误识为“w”；“绕”和“让”有相同的声母“r”和声调“4”，语音识别错误来源于韵母“ao”被误识为“ang”。示例 3 中，“院方协商”与误识的“岳风学生”有着相同的声母和声调，其语音识别错误来源于“院方协商”的韵母“uan”、“ang”、“ie”、“ang”分别被误识为“ue”、“eng”、“ue”、“eng”。由此看来，语音识别文本错误表现为语句中某些字符的拼音组成部分发生了变化，这些字符被误识为与其有相同声母或韵母的字符。

3.2 基于 RPIF 的数据增强方法

根据 ASR 系统识别错误表现为语句中的某些字符被误识为其同声母或同韵母字符的特点，本文提出一种基于 RPIF 的数据增强方法，如算法 1 所示。在此之前，给出以下定义。

定义 6 同声字符集。设字符集 $C_{\mathrm{i}}=\{c_1,\cdots,c_n\}$，$n\in Z$，若 C_{i} 中字符的声母都相同，则称 C_{i} 为同声字符集。

定义 7 同韵字符集。设字符集 $C_{\mathrm{f}}=\{c_1,\cdots,c_n\}$，$n\in Z$，若 C_{f} 中字符的韵母都相同，则称 C_{f} 为同韵字符集。

定义 8 同声字典。多个声母及其同声字符集构成的集合。

定义 9 同韵字典。多个韵母及其同韵字符集构成的集合。

定义 10 声韵混淆集。一个汉字对应一个声韵混淆集，声韵混淆集中任意一个字符都与该汉字有相同的声母或韵母。

算法 1 基于 RPIF 的数据增强方法

输入 源语料库 C_{s}，汉字集 C_{c}，单条语句的最大错误字符个数 $n_{\max}$，目标语料库大小 N，置换概率 P，同声字典 D_{i}，同韵字典 D_{f}

输出 目标语料库 C_{e}

$D_{\mathrm{i}}\leftarrow\varnothing$，$D_{\mathrm{f}}\leftarrow\varnothing$，$C_{\mathrm{e}}\leftarrow\varnothing$
for x in C_{c} do
  获取x的声母 x_{i}，根据 x_{i} 将x加入 D_{i}
  获取x的韵母 x_{f}，根据 x_{f} 将x加入 D_{f}
end for
for i in range (N) do
  $j=i \bmod \mathrm{len}(C_{\mathrm{s}})$ //$\mathrm{len}(C_{\mathrm{s}})$ 指源语料库大小
  $(S_{\mathrm{s}},S_{\mathrm{l}})\leftarrow C_{\mathrm{s}}[j]$ //源语料库的第j个语句对
  $r\leftarrow\mathrm{rand}(0,1)$ //0～1的随机数
  if $r>P$ then
    将 $(S_{\mathrm{s}},S_{\mathrm{l}})$ 添加到 C_{e}
  else
    $S_{\mathrm{e}}\leftarrow S_{\mathrm{l}}$ // S_{e} 指生成的错误语句
    $n\leftarrow\mathrm{randint}(1,n_{\max})$ //1～$n_{\max}$ 的随机整数
    if $n\geqslant\mathrm{len}(S_{\mathrm{l}})$ then // $\mathrm{len}(S_{\mathrm{l}})$ 指 S_{l} 的长度
      chars $\leftarrow S_{\mathrm{l}}$
    else
      chars $\leftarrow$ 随机抽取 S_{l} 的n个字符
    end if
    for c in chars do
      获取c的声母 c_{i} 和韵母 c_{f}
      根据 c_{i} 从 D_{i} 中查询c的同声字符集 C_{i}
      根据 c_{f} 从 D_{f} 中查询c的同韵字符集 C_{f}
      $C_{\mathrm{mix}}\leftarrow\{C_{\mathrm{i}}\cup C_{\mathrm{f}}\}$ //c的声韵混淆集
      $c_{\mathrm{alter}}\leftarrow$ 随机抽取 C_{mix} 中的一个字符
      将 S_{e} 中的c置换为 c_{alter}
    end for
    将 $(S_{\mathrm{e}},S_{\mathrm{l}})$ 添加到 C_{e}
  end if
end for
return C_{e}

表 2 **Kaldi 语音识别工具包使用过程中出现的错误示例**

示例	语音识别错误语句	错误字符的拼音	相应的正确语句	相应正确字符的拼音
1	但是不**行**最终还是发生了	xing**2**	但是不**幸**最终还是发生了	xing**4**
2	**没让**亚运会进行的城市资金投入	**m**ei2/ r**ang**4	**围绕**亚运会进行的城市资金投入	**w**ei2/r**ao**4
3	与**岳风学生**赔偿问题	y**ue**4/f**eng**1/x**ue**2/sh**eng**1	与**院方协商**赔偿问题	y**uan**4/f**ang**1/x**ie**2/sh**ang**1

算法 1 展示了基于 RPIF 的数据增强方法的详细过程，该过程主要是将从语句中随机抽取的 n 个字符分别置换为与其同声母或同韵母字符的方式来获取大量的纠错语料。算法 1 中的置换概率 P 决定了目标语料库中生成语料与源语料的比例，生成语料随 P 的增大而增多。当 $P=0$ 时，目标语料库的数据是对源语料库的复制扩充。当 $P=1$ 时，目标语料库的数据均是采用算法 1 中步骤 13)～步骤 28)所示方法获取的生成语料。此时，目标语料库 C_e 的可扩展规模受汉字集 C_c 大小的影响。C_c 越大，单个汉字的声韵混淆集越大，纠错语料库的上限规模便会越大。值得注意的是，算法 1 中的步骤 14)、步骤 18)和步骤 25)均采用随机化方式来设置当前语句的错误字符个数 n、抽取 n 个待替换字符及其替换字符，这能有效地模拟 ASR 系统识别错误出现的随机性。

4　实验

本节首先介绍实验所用数据集、实验环境和评估指标。然后将基于 CSPI 的文本校对模型和 2 个未结合拼音编码方法的模型进行比较，以验证基于 CSPI 的文本校对模型的检错和纠错性能，并对比不同拼音编码方法对模型性能的影响。最后设置 2 组实验分别验证优化目标和基于 RPIF 的数据增强方法对基于 CSPI 的模型校对性能的影响。

4.1　实验数据

为了验证音韵信息对语音识别文本校对任务的影响，本文使用多人普通话语音数据集 AISHELL-3[38]开展了相关实验，并使用 Kaldi[39]作为语音识别工具来获取该数据集的识别文本。AISHELL-3 数据集共包含 88 035 条数据，其中训练集为 63 262 条，测试集为 24 773 条。由于本文面向语音识别后替换错误的文本校对，在获取语音识别文本后筛选出无重复且与标签文本长度相等的语句作为实验数据。最终得到 34 899 条实验数据，其中训练集为 24 813 条，测试集为 9 888 条。详细的实验数据统计信息如表 3 所示。表 3 中，Train 表示训练集，Test 表示测试集，Total 表示数据总数，True 表示正确语句条数，False 表示错误语句条数，Error 表示语句中错误的字符个数，Len 表示语句包含的字符个数。

表 3　AISHELL-3 数据集实验数据统计信息

Data/条	Total/个	True/个	False/个	Len/个	Error/个	语句占比	
						Len<5	Len<10
Train	24 813	300	24 513	2～39	1～15	6.11%	43.65%
Test	9 888	104	9 784	1～39	1～16	12.91%	49.61%

4.2　实验环境及模型评估

本文实验环境如下：操作系统为 64 位 Windows10 系统，CPU 为英特尔 i9-10850K，GPU 为 16 GB 的 NVIDIA A4000，内存为 DDR4 32 GB。实验中涉及的深度学习模型使用 Pytorch 构建。训练模型的参数设置如表 4 所示。在模型训练过程中，从训练集中随机抽取 20%的数据作为验证集。

表 4　训练模型的参数设置

参数	值
迭代轮次	150
批量大小	32
优化器	Adam
学习率	0.001
丢弃率	0.2
卷积核大小	2
编码器和解码器的层数	1
嵌入向量的维度	256
编码器隐藏向量的维度	128
解码器隐藏向量的维度	128

为客观评估模型性能，取模型在 AISHELL-3 数据集上 5 次实验结果的均值作为最终的模型性能评估数据，选用文本纠错任务中常用的准确率（P, precision）、召回率（R, recall）、F1（F1-measure）作为评估指标[23]，并主要通过 F1 值来对比不同模型的检错和纠错性能。

4.3　实验结果与分析

4.3.1　拼音编码方法的有效性

本节将基于 CSPI 的文本校对模型与以下 2 个无拼音编码模型进行比较，以此检验拼音编码方法的有效性。同时，通过对比不同拼音编码模型的检错和纠错结果，验证不同拼音编码方法对模型性能的影响。无拼音编码模型简介如下。

1) M_C[24]。使用 2 层 1D-CNN 和注意力机制构建的基于编码器–解码器架构的文本校对模型。模型参数与表 4 所列各项参数保持一致。

2) M_G。使用门控循环单元（GRU, gated recurrent unit）和注意力机制构建基于编码器–解码器架构的文本校对模型，即图 2 所示模型仅以字符作为模型输入。

为了便于说明，将基于 CSPI 的文本校对模型使用 P_U、P_B、P_C、P_{CU} 和 P_{CB} 这 5 种拼音编码方法时分别记为 M_G+P_U、M_G+P_B、M_G+P_C、M_G+P_{CU} 和 M_G+P_{CB}，统称为拼音编码模型 M_G+P。各拼音编码模型和无拼音编码模型的检错和纠错结果如图 5 和表 5 所示。

由图 5 和表 5 可以看出，各个拼音编码模型的检错结果均显著优于无拼音编码模型，同时拼音编码模型的纠错结果也优于无拼音编码模型。对比 2 种无拼音编码模型，M_G 的检错和纠错结果始终优于 M_C。接下来，从检错和纠错 2 个方面详细地分析各个模型的文本校对性能。

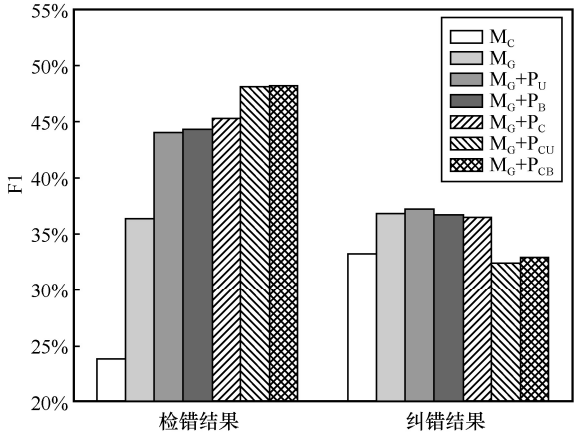

图 5 拼音编码模型和无拼音编码模型的文本校对性能对比

由图 5 和表 5 可以看出，混合网络拼音编码模型 M_G+P_{CB} 的检错 F1 值优于 M_G+P_{CU}，且两者的检错结果明显优于单网络拼音编码模型。对比单网络拼音编码模型的检错 F1 值可以看出，M_G+P_C 优于 M_G+P_B，M_G+P_B 优于 M_G+P_U。具体来说，M_G+P_{CB} 取得了最高检错 F1 值 48.16%，相较 M_G 和 M_C 分别高出 11.91%和 24.31%，相较 M_G+P_U、M_G+P_B、M_G+P_C 和 M_G+P_{CU} 分别高出 4.13%、3.82%、2.94%和 0.13%。这与本文预期的效果相同，复杂的拼音编码网络能促使模型提取分辨能力较强的音韵信息，有助于模型检测文本错误。此外，由图 5 和表 5 还可以看出，拼音编码模型的检错准确率随拼音编码网络复杂度的增加而降低，但其检错召回率和 F1 值随拼音编码网络复杂度的增加而不断增大，模型检错性能整体向好。这说明基于 CSPI 的文本校对模型结合复杂度较高的拼音编码网络可以增强其检测错误字符的灵敏度，进而增加真实错误字符的检出率。

由表 5 可得，拼音编码模型的各项纠错指标有随拼音编码网络复杂度的增加而下降的趋势。对比各个模型的纠错 F1 值，拼音编码模型 M_G+P_U 取得了最高纠错 F1 值 37.21%，比无拼音编码模型 M_G 和 M_C 分别高出 0.43%和 3.98%。而其他拼音编码模型的纠错性能却低于无拼音编码模型，且混合网络拼音编码模型的纠错性能不如单网络拼音编码模型。拼音编码模型的纠错性能整体呈现与其检错性能相反的趋势。这是因为中文存在较多同音异义的字符，模型使用复杂的拼音编码方法提取的音韵信息分辨能力过强，导致模型认为原有错误字符或模型预测的字符在语音或语义上都能使句子有意义，本文将此称为由音韵信息引起的过纠现象。

综上所述，音韵信息有利于基于 CSPI 的文本校对模型检测和纠正文本错误。模型的检错能力随拼音编码网络的复杂度增加而增强。由于存在音韵信息引起的过纠现象，模型的纠错能力呈现随拼音编码网络的复杂度增加而下降的趋势。

表 5 **拼音编码模型和无拼音编码模型的文本校对性能对比结果**

模型	检错结果			纠错结果		
	P	R	F1	P	R	F1
M_C	59.53%	14.93%	23.85%	48.45%	25.29%	33.23%
M_G	56.76%	26.63%	36.25%	**50.24%**	29.01%	36.78%
M_G+P_U	**63.58%**	33.71%	44.03%	48.58%	**30.16%**	**37.21%**
M_G+P_B	63.18%	34.19%	44.34%	47.78%	29.77%	36.68%
M_G+P_C	61.63%	35.72%	45.22%	47.49%	29.67%	36.52%
M_G+P_{CU}	48.29%	**48.24%**	48.03%	40.07%	27.11%	32.33%
M_G+P_{CB}	49.69%	46.92%	**48.16%**	41.00%	27.47%	32.90%

4.3.2　优化目标对模型性能的影响

本节主要通过对比不同拼音编码模型使用字符优化目标 $\mathcal{L}_c$ 和字符–拼音优化目标 $\mathcal{L}_{cp}$ 时的检错和纠错结果来分析优化目标对模型性能的影响。各模型的文本校对结果如图 6 和表 6 所示。接下来，从检错和纠错 2 个方面对比分析各个模型的文本校对性能。

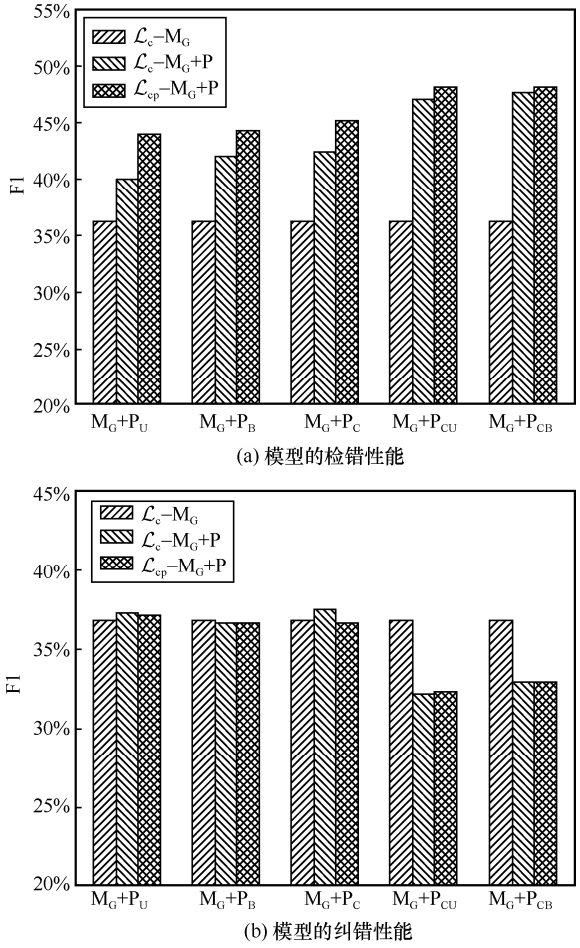

图 6　基于 CSPI 的模型使用不同优化目标时的文本校对性能对比

由表 6 和图 6(a)可以看出，拼音编码模型 M_G+P 无论使用 $\mathcal{L}_c$ 还是 $\mathcal{L}_{cp}$，其检错性能均优于无拼音编码模型 M_G。相较使用 $\mathcal{L}_c$，M_G+P_U、M_G+P_B、M_G+P_C、M_G+P_{CU} 和 M_G+P_{CB} 使用 $\mathcal{L}_{cp}$ 时的检错 F1 值分别提升了 4.13%、2.34%、2.83%、0.97%和 0.42%，这说明字符–拼音优化目标能够促使模型学习分辨能力更强的音韵信息，进而提升了模型的检错性能。由表 6 和图 6(a)还可以看出，当模型使用 $\mathcal{L}_c$ 时，M_G+P_{CB} 的检错 F1 值比 M_G+P_{CU} 高，且两者的检错性能仍明显优于 M_G+P_U、M_G+P_B 和 M_G+P_C，这也进一步体现了模型融合复杂拼音编码网络学习的音韵信息更加有利于其辨别文本错误。

然而，由表 6 和图 6(b)可以看出，M_G+P_C 使用 $\mathcal{L}_c$ 时的纠错结果高于其使用 $\mathcal{L}_{cp}$，此时拼音编码模型取得了最优纠错 F1 值 37.46%，相较 M_G+P_C 使用 $\mathcal{L}_{cp}$ 的纠错 F1 值高出 0.94%，相较 M_G 的纠错 F1 值高出 0.68%，相较 M_G+P_U 使用 $\mathcal{L}_{cp}$ 取得的最好纠错 F1 值高出 0.25%。M_G+P_U、M_G+P_B 和 M_G+P_{CB} 使用 $\mathcal{L}_c$ 和 $\mathcal{L}_{cp}$ 时取得的纠错结果相当。仅 M_G+P_{CU} 使用 $\mathcal{L}_{cp}$ 时的纠错结果优于其使用 $\mathcal{L}_c$。

综上所述，在使用字符–拼音优化目标时，基于 CSPI 的文本校对模型结合复杂拼音编码网络提取的音韵信息能够使其具备更好的文本错误检测能力。在使用字符优化目标时，基于 CSPI 的文本校对模型结合简单拼音编码网络提取的音韵信息使其纠错能力占有一定的优势。

4.3.3　基于 RPIF 的数据增强方法的影响

根据以上实验结果，本节选取单网络拼音编码模型 M_G+P_C 和混合网络拼音编码模型 M_G+P_{CB} 来验证基于 RPIF 的数据增强方法对模型检错和纠错性能的影响。

表 6　　**基于 CSPI 的模型使用不同优化目标时的文本校对性能对比结果**

模型	检错结果						纠错结果					
	P		R		F1		P		R		F1	
	$\mathcal{L}_c$	$\mathcal{L}_{cp}$	$\mathcal{L}_c$	$\mathcal{L}_{cp}$	$\mathcal{L}_c$	$\mathcal{L}_{cp}$	$\mathcal{L}_c$	$\mathcal{L}_{cp}$	$\mathcal{L}_c$	$\mathcal{L}_{cp}$	$\mathcal{L}_c$	$\mathcal{L}_{cp}$
M_G	56.76%	—	26.63%	—	36.25%	—	**50.24%**	—	29.01%	—	36.78%	—
M_G+P_U	**63.84%**	**63.58%**	29.07%	33.71%	39.90%	44.03%	49.93%	**48.58%**	29.69%	**30.16%**	37.24%	**37.21%**
M_G+P_B	62.76%	63.18%	31.58%	34.19%	42.00%	44.34%	48.62%	47.78%	29.44%	29.77%	36.67%	36.68%
M_G+P_C	62.20%	61.63%	32.17%	35.72%	42.39%	45.22%	49.43%	47.49%	**30.17%**	29.67%	**37.46%**	36.52%
M_G+P_{CU}	46.75%	48.29%	48.12%	**48.24%**	47.06%	48.03%	39.77%	40.07%	26.97%	27.11%	32.14%	32.33%
M_G+P_{CB}	46.31%	49.69%	**49.52%**	46.92%	**47.74%**	**48.16%**	40.49%	41.00%	27.73%	27.47%	32.91%	32.90%

算法 1 所需输入参数如下。源语料库 C_s 为 AISHELL-3 的训练集。汉字集 C_c 选用《通用规范汉字字典》[40]的一级字表和二级字表，共包含 6 500 个常用汉字。单条语句的最大错误字符个数 $n_{\max}=4$，置换概率为 $P=1$。目标语料库大小 N 分别设置为 100 000、150 000 和 200 000，记为 10w、15w 和 20w。M_G+P_C 和 M_G+P_{CB} 使用不同大小目标语料库训练时的文本校对性能对比结果如表 7 所示。表 7 中，Origin 表示模型训练集为原始训练集大小。

由表 7 可以看出，M_G+P_C 和 M_G+P_{CB} 的检错召回率和 F1 值随着目标语料库的增大而增大，其检错准确率也随目标语料库的增大有不同程度的提升。当训练集大小为 20w 时，M_G+P_C 和 M_G+P_{CB} 取得了最优检错 F1 值，分别为 49.57%和 51.20%，相较使用原始训练集，其检错 F1 值分别提升了 4.35%和 3.04%。这表明由基于 RPIF 的数据增强方法获取的训练集能促使模型学习更多文本错误实例的音韵信息，进而加强了模型检测文本错误的能力。由表 7 还可以看出，当模型使用同一语料库训练时，M_G+P_{CB} 的检错结果始终优于 M_G+P_C，这进一步验证了基于 CSPI 的文本校对模型所结合的拼音编码网络的复杂度越高，其检错能力越好。

由表 7 也可以看出，M_G+P_C 和 M_G+P_{CB} 的纠错结果并未随着目标语料库的增大而增大。这是由于训练集中的混淆音错误字符随数据量增加而不断增多，模型学习的语义信息受到了影响。此外，从表 7 还可以看出，M_G+P_C 的各项纠错指标优于 M_G+P_{CB}，这与表 5 和表 6 所反映的信息一致，基于 CSPI 的文本校对模型结合简单拼音编码网络学习的音韵信息更有助于其纠正文本错误。

5 讨论

拼音携带的音韵信息有助于文本校对模型检测和纠正语音识别后的文本错误，这与文献[7,9]得出的结论一致。结合表 5～表 7 可以看出，基于 CSPI 的文本校对模型取得的最优检错 F1 值比无拼音编码模型 M_C 和 M_G 分别高 27.35%和 14.95%；其最优纠错 F1 值比 M_C 和 M_G 分别高 4.23%和 0.68%。由表 5～表 7 所示实验结果还可以看出，模型结合复杂拼音编码网络提取的音韵信息更有利于其检出文本错误，但模型的纠错性能会受到影响。本文认为这是一种音韵信息引起的过纠现象。模型结合复杂拼音编码网络能够提取到分辨力较强的音韵信息，进而提升了检测文本错误的灵敏度。但音韵信息过强会导致模型认为某些错误字符也能使句子在语音或语义上有意义，以致模型无法纠正此类文本错误。文献[30]中也提及了类似的问题。例如，“的”、“地”和“得”3 个字有相同的发音“de”，将语句中“地”替换为其他两者后，该语句依然有意义。

加大拼音编码网络的复杂度、加强模型训练过程中对音韵信息的优化、增加训练数据中混淆音文本错误的类别均能促使文本校对模型捕获较强分辨力的音韵信息，进而提升模型的文本检错能力。降低拼音编码网络的复杂度或在模型训练过程中适当减少对音韵信息的优化则有利于文本校对模型纠正文本错误。由表 6 可以看出，基于 CSPI 的文本校对模型结合任意一种拼音编码网络且使用字符−拼音优化目标时都能取得更好的检错性能；而当仅使用字符优化目标时，模型的纠错性能更好。这是由于仅使用字符优化目标能够在一定程度上削弱音韵信息引起的过纠现象。由表 7 可以看出，基于 CSPI 的文本校对模型结合复杂拼音编码网络

表 7　　基于 CSPI 的模型使用不同大小目标语料库训练时的文本校对性能对比结果

数据大小	检错结果						纠错结果					
	P		R		F1		P		R		F1	
	M_G+P_C	M_G+P_{CB}	M_G+P_C	M_G+P_{CB}	M_G+P_C	M_G+P_{CB}	M_G+P_C	M_G+P_{CB}	M_G+P_C	M_G+P_{CB}	M_G+P_C	M_G+P_{CB}
Origin	**61.63%**	49.69%	35.72%	46.92%	45.22%	48.16%	**47.49%**	**41.00%**	**29.67%**	**27.47%**	**36.52%**	**32.90%**
10w	59.41%	49.39%	37.55%	49.27%	46.02%	49.33%	41.21%	33.61%	26.20%	23.28%	32.03%	27.51%
15w	56.71%	52.19%	42.57%	49.94%	48.63%	51.04%	37.22%	35.48%	24.43%	24.73%	29.50%	29.15%
20w	56.80%	**52.27%**	**43.97%**	**50.18%**	**49.57%**	**51.20%**	37.96%	34.56%	25.19%	24.07%	30.28%	28.38%

且使用字符–拼音优化目标时，其检错性能随训练集中混淆音文本错误的增加有进一步提升。综上所述，本文建议借助音韵信息校对语音识别文本错误时，分开进行检错与纠错这 2 个子任务，通过融合复杂拼音编码网络并在训练过程中加强对音韵信息的优化来提升文本校对模型的检错率，通过融合简单拼音编码网络或在训练过程中适当减少对音韵信息的优化来辅助提升文本校对模型的纠错率。

文本长度较短及上下文语义缺失是语音识别文本校对任务的难点。由表 5～表 7 可以看出，各类模型的文本校对性能一般。本文认为这主要是由于来自 ASR 系统的文本长度较短，模型很难根据句子的上下文语义来纠错。例如，“吃饭了吗”容易因用户发音不清晰被 ASR 系统误识为“吃饭了啊”。若不考虑语境，可以认为后者是正确的，由此可见，模型校对此类短文本的难度较高。由表 3 可知，AISHELL 测试集中长度小于 5 和小于 10 的语句分别占 12.91%和 49.61%。此外，由表 7 可知，当使用基于 RPIF 的数据增强方法扩充模型的训练集后，模型的检错性能随着训练数据的逐步增加而不断提升，但其纠错性能却呈现随着训练数据的增加而降低的趋势，可能的原因有 2 个，一个是训练数据中混淆音错误字符的增多加重了由音韵信息引起的过纠现象；另一个是本文用于验证拼音编码方法的文本校对模型的结构相对简单，模型学习语义信息的能力受限。在今后的工作中，尝试将大规模的预训练语言模型和拼音编码方法相结合来解决语音识别后的文本校对问题。

6　结束语

本文提出了 P_U、P_B、P_C、P_{CU} 和 P_{CB} 这 5 种拼音编码方法，并以此构建了基于 CSPI 的文本校对模型，实现了同时利用句子的语义和音韵信息校对语音识别文本错误。针对标注数据有限造成许多先进的深度学习模型难以应用于语音识别文本校对任务的问题，本文提出了一种基于 RPIF 的数据增强方法。在多人普通话语音数据集 AISHELL-3 上进行了相关实验，实验结果表明，拼音携带的音韵信息有利于文本校对模型检测和纠正语音识别文本错误。基于 CSPI 的文本校对模型使用混合网络拼音编码方法（P_{CU}、P_{CB}）所提取的音韵信息有利于其检测语音识别文本错误，使用单网络拼音编码方法（P_U、P_B、P_C）所提取的音韵信息则更利于其纠正语音识别文本错误。所提数据增强方法能促使文本校对模型学习更多语音识别错误实例，有效地提升了模型检出语音识别文本错误的能力。在未来的研究工作中，笔者会尝试将预训练语言模型与不同的拼音编码方法相结合，分别用于语音识别文本错误的检测和纠正，以进一步辅助 ASR 系统提升其识别准确性。

参考文献：

[1] ERRATTAHI R, HANNANI A E, OUAHMANE H. Automatic speech recognition errors detection and correction: a review[J]. Procedia Computer Science, 2018, 128: 32-37.

[2] ZHANG S L, LEI M, YAN Z J. Investigation of transformer based spelling correction model for CTC-based end-to-end mandarin speech recognition[C]//Proceedings of the International Speech Communication Association (INTERSPEECH). Grenoble: International Speech Communication Association, 2019: 2180-2184.

[3] ZHAO Y, YANG X R, WANG J C, et al. BART based semantic correction for Mandarin automatic speech recognition system[C]//Proceedings of the International Speech Communication Association (INTERSPEECH). Grenoble: International Speech Communication Association, 2021: 2017-2021.

[4] WANG X Q, LIU Y Q, ZHAO S, et al. A light-weight contextual spelling correction model for customizing transducer-based speech recognition systems[C]//Proceedings of the International Speech Communication Association (INTERSPEECH). Grenoble: International Speech Communication Association, 2021: 1982-1986.

[5] ZHANG S L, LEI M, LIU Y, et al. Investigation of modeling units for mandarin speech recognition using dfsmn-ctc-smbr[C]//Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE Press, 2019: 7085-7089.

[6] YANG L, LI Y, WANG J, et al. Post text processing of Chinese speech recognition based on bidirectional LSTM networks and CRF[J]. Electronics, 2019, 8(11): 1248.

[7] CHEN Y C, CHENG C Y, CHEN C A, et al. Integrated semantic and phonetic post-correction for Chinese speech recognition[C]//Proceedings of Conference on Computational Linguistics and Speech Processing (ROCLING). Stroudsburg: Association for Computational Linguistics, 2021: 95-102.

[8] LI M, DANILEVSKY M, NOEMAN S, et al. DIMSIM: an accurate Chinese phonetic similarity algorithm based on learned high dimensional encoding[C]//Proceedings of the 22nd Conference on Computational Natural Language Learning. Stroudsburg: Association for Computational Linguistics, 2018: 444-453.

[9] DUAN D G, LIANG S H, HAN Z M, et al. Pinyin as a feature of neural machine translation for Chinese speech recognition error correction[C]//China National Conference on Chinese Computational Linguistics (CCL). Berlin: Springer, 2019: 651-663.

[10] JIANG Y, WANG T, LIN T, et al. A rule based Chinese spelling and grammar detection system utility[C]//Proceedings of 2012 International Conference on System Science and Engineering (ICSSE). Piscata-

way: IEEE Press, 2012: 437-440.

[11] CHU W C, LIN C J. NTOU Chinese spelling check system in SIGHAN-8 bake-off[C]//Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 102-107.

[12] XU H D, LI Z L, ZHOU Q Y, et al. Read, listen, and see: leveraging multimodal information helps Chinese spell checking[C]//Proceedings of Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Stroudsburg: Association for Computational Linguistics, 2021: 716-728.

[13] 王辰成, 杨麟儿, 王莹莹, 等. 基于 Transformer 增强架构的中文语法纠错方法[J]. 中文信息学报, 2020, 34(6): 106-114.
WANG C C, YANG L E, WANG Y Y, et al. Chinese grammatical error correction method based on transformer enhanced architecture[J]. Journal of Chinese Information Processing, 2020, 34(6): 106-114.

[14] 段建勇, 袁阳, 王昊. 基于Transformer局部信息及语法增强架构的中文拼写纠错方法[J]. 北京大学学报(自然科学版), 2021, 57(1): 61-67.
DUAN J Y, YUAN Y, WANG H. Chinese spelling correction method based on transformer local information and syntax enhancement architecture[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2021, 57(1): 61-67.

[15] ZHUANG L, BAO T, ZHU X, et al. A Chinese OCR spelling check approach based on statistical language models[C]//Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics. Piscataway: IEEE Press, 2004: 4727-4732.

[16] XIE W J, HUANG P J, ZHANG X R, et al. Chinese spelling check system based on N-gram model[C]//Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 128-136.

[17] LIU X D, CHENG F, DUH K, et al. A hybrid ranking approach to Chinese spelling check[J]. ACM Transactions on Asian and Low-Resource Language Information Processing, 2015, 14(4): 1-17.

[18] 冯海林, 张潇, 刘同存. 融合评论文本特征和评分图卷积表示的推荐模型[J]. 通信学报, 2022, 43(3): 164-171.
FENG H L, ZHANG X, LIU T C. Recommendation model combining review's feature and rating graph convolutional representation[J]. Journal on Communications, 2022, 43(3): 164-171.

[19] 张煜, 吕锡香, 邹宇聪, 等. 基于生成对抗网络的文本序列数据集脱敏[J]. 网络与信息安全学报, 2020, 6(4): 109-119.
ZHANG Y, LYU X X, ZOU Y C, et al. Differentially private sequence generative adversarial networks for data privacy masking[J]. Chinese Journal of Network and Information Security, 2020, 6(4): 109-119.

[20] 叶俊民, 罗达雄, 陈曙. 基于层次化修正框架的文本纠错模型[J]. 电子学报, 2021, 49(2): 401-407.
YE J M, LUO D X, CHEN S. A text error correction model based on hierarchical editing framework[J]. Acta Electronica Sinica, 2021, 49(2): 401-407.

[21] 郭可翔, 王衡军, 白祉旭. 融合多通道 CNN 与 BiGRU 的字词级文本错误检测模型[J]. 计算机工程, 2022, 48(9): 63-70.
GUO K X, WANG H J, BAI Z X. Detection model for word-level text error combining multi-channel CNN and BiGRU[J]. Computer Engineering, 2022, 48(9): 63-70.

[22] WANG D M, 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. Stroudsburg: Association for Computational Linguistics, 2018: 2517-2527.

[23] WANG D M, 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. Stroudsburg: Association for Computational Linguistics, 2019: 5780-5785.

[24] CHOLLAMPATT S, NG H T. A multilayer convolutional encoder-decoder neural network for grammatical error correction[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2018: 5755-5762.

[25] LIU C L, LAI M H, TIEN K W, et al. Visually and phonologically similar characters in incorrect Chinese words[J]. ACM Transactions on Asian Language Information Processing, 2011, 10(2): 1-39.

[26] WANG H, WANG B, DUAN J Y, et al. Chinese spelling error detection using a fusion lattice LSTM[J]. ACM Transactions on Asian and Low-Resource Language Information Processing, 2021, 20(2): 1-11.

[27] LIU S L, YANG T, YUE T C, 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 (Volume 1: Long Papers). Stroudsburg: Association for Computational Linguistics, 2021: 2991-3000.

[28] HONG Y Z, YU X G, HE N, et al. 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). Stroudsburg: Association for Computational Linguistics, 2019: 160-169.

[29] ZHANG S H, HUANG H R, LIU J C, et al. Spelling error correction with soft-masked BERT[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 2020: 882-890.

[30] CHENG X Y, XU W D, CHEN K L, 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. Stroudsburg: Association for Computational Linguistics, 2020: 871-881.

[31] JI T, YAN H, QIU X P. SpellBERT: a lightweight pretrained model for Chinese spelling check[C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2021: 3544-3551.

[32] ZHANG R Q, PANG C, ZHANG C Q, et al. Correcting Chinese spelling errors with phonetic pre-training[C]//Proceedings of Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Stroudsburg: Association for Computational Linguistics, 2021: 2250-2261.

[33] TSENG Y H, LEE L H, CHANG L P, et al. Introduction to SIGHAN 2015 bake-off for Chinese spelling check[C]//Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 32-37.

[34] CHO K, MERRIENBOER B V, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg: Association for Computational Linguistics, 2014: 1724-1734.

[35] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Annual Conference on Neural Information Processing Systems (NeurIPS). Cambridge: MIT Press, 2014: 3104-3112.

[36] GRUNDKIEWICZ R, JUNCZYS-DOWMUNT M. Near human-level performance in grammatical error correction with hybrid machine translation[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Stroudsburg: Association for Computational Linguistics, 2018: 284-290.

[37] LUONG T, PHAM H, MANNING C D. Effective approaches to attention-based neural machine translation[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Stroudsburg: Association for Computational Linguistics, 2015: 1412-1421.

[38] SHI Y, BU H, XU X, et al. AISHELL-3: a multi-speaker mandarin TTS corpus and the baselines[J]. arXiv Preprint, arXiv: 2010.11567, 2020.

[39] POVEY D, GHOSHAL A, BOULIANNE G, et al. The Kaldi speech recognition toolkit[C]//IEEE Workshop on Automatic Speech Recognition and Understanding (CONF). Piscataway: IEEE Press, 2011: 1-4.

[40] 王宁. 通用规范汉字字典[M]. 北京: 商务印书馆, 2013.
WANG N. The general specification Chinese character dictionary[M]. Beijing: The Commercial Press, 2013.

[作者简介]

仲美玉（1993− ），女，河北邢台人，燕山大学博士生，主要研究方向为智能信息处理。

吴培良（1981− ），男，河北石家庄人，博士，燕山大学教授、博士生导师，主要研究方向为自然语言处理、深度强化学习、机器人操作技能学习。

窦燕（1968− ），女，陕西西安人，博士，燕山大学教授、硕士生导师，主要研究方向为智能信息处理、机器视觉与模式识别。

刘毅（1998− ），男，河北石家庄人，燕山大学硕士生，主要研究方向为智能信息处理、机器视觉。

孔令富（1957− ），男，吉林公主岭人，博士，燕山大学教授、博士生导师，主要研究方向为智能控制与智能信息处理、机器人视觉。