CCL2022-CLTC赛道一:中文拼写检查

马延美　陈政华　付钰楚　张杰
北京方寸无忧科技发展有限公司
{mayanmei, chenzhenghua, fuyuchu, zhangjie}@ifuncun.cn

摘要

中文拼写检查任务目的是检测并纠正中文文本中的拼写错误，CCL2022-CLTC赛道一的拼写错误包括：音近、形近、形音兼近三种。我们在baseline基础上，先后进行了数据扩充、数据增强、模型调研和模型微调，以及引入混淆集、困惑度等策略，最终使得模型效果在测试集上的句子级别的C-F达到83.19。

关键词： 中文拼写检查；音近；形近；音形兼近

The Track1 of The Twenty-first China National Conference on Computational Linguistics (CCL-2022): Chinese spelling check

Yanmei Ma, Zhenghua Chen, Yuchu Fu, Jie Zhang
Beijing Funcun-wuyou Technology Co., Ltd.
{mayanmei, chenzhenghua, fuyuchu, zhangjie}@ifuncun.cn

Abstract

Finding and fixing spelling mistakes in Chinese text is the goal of the assignment for Chinese spelling check. Track 1 assignments include morphological close, phonetic close, and both morphological and phonological close. Using the baseline as a starting point, we perform data enlargement, data enhancement, model research, model fine-tuning, as well as introduce techniques like confusion set and perplexity, until the model influence on the test set reaches 0.8319 at the sentence level.

Keywords: Chinese spelling check , phonetic close , morphological close

1 引言

很多自然语言处理落地场景都会涉及到文本纠错的相关技术，例如跟各种形式机器人的语音或者文字对话，或者用手机扫描相关的PDF或者图片，或者跟人聊天时用输入法打字等等，无论是通过ASR识别的语音信息，通过OCR识别得到的图片信息，还是用户真实通过输入法的文字，都有可能出现错误。这些错误会影响文本的可读性，不利于人和机器的理解，如果这些错误不加处理，会传播到后续的环节，影响后续任务的效果。

中文拼写检查一般包括检错和纠错两部分，检查和纠正正文中的拼写错误。常见的文本错误类型包括：同音字、音近字、形近字、字词颠倒、多字、少字、拼音全拼或缩写等。CCL2022-CLTC(Wang et al., 2022)赛道一的拼写错误包括：音近、形近、形音兼近三种。
赛道一提供的是基于Bert的基线模型，使用的预训练语言模型是bert-base-chinese，我们先用赛道一提供的170条验证集做训练集，在基线模型上做微调，在赛道一测试集上进行测试，测试结果句子级别的C-F为49.34。然后我们做了数据扩充，加入比赛提供的SIGHAN和Wang271k（因Lang8的错误类型和赛道一任务不同，没有使用）合成数据集27万，切分训练集22万，验证集5万，再在基线模型上进行微调，赛道一测试集结果句子级别的C-F为：67.59。
我们又在22万训练集上做了数据增强，增强后的训练集为36万，在基线模型上进行微调。同时做了大量的模型调研，包括：ReaLiSe (Xu et al., 2021)、CRASpell (Liu et al., 2022)、Macbert4csc (Cui et al., 2020)、SpellGCN (Cheng et al., 2020)等模型。我们对比分析分别用这些模型，先后在增强前的22万和增强后的36万数据集上进行训练，在赛道一测试集上进行预测，得到不同的测试结果，对比不同模型的纠错能力，在确定错误位置时取并集，每个模型的纠正结果加入候选集，引入先验知识过滤掉一些误报错误，通过N-gram计算困惑度的方法来提高纠错准确率。
本次实验的所有代码已上传到github，代码链接如下：https://github.com/HuaC-Z/CCL2022-track1

2 模型

因为我们在做赛道一任务的时候做了大量的模型和论文调研，主要模型包括：ReaLiSe (Xu et al., 2021)、CRASpell (Liu et al., 2022)、Macbert4csc (Cui et al., 2020)、SpellGCN (Cheng et al., 2020)等。其中，SpellGCN只做纠错，ReaLiSe、CRASpell、Macbert4csc是检错纠错一起。在这里对我们用到的模型做个简单介绍。

2.1 ReaLiSe

ReaLiSe (Xu et al., 2021)算法使用文本、声音、视觉三个编码器学习信息表示，然后选择性模态融合模块来获得上下文感知的多模态表示，最后输出预测错误纠正的概率。算法细节如下：
1. 算法采用bert作为语义编码器的主干来捕获文本信息，用BERT-wwm模型的权重初始化语义编码器，语义编码器的架构与BERTBASE模型相同（12个注意头，12个Transformer层，隐层大小为768）；
2. 对于声音形态，使用汉语拼音作为特征，使用分层编码器处理字符级和句子拼音字母，语音句级编码器，我们将层数设置为4层，并用BERT的位置嵌入初始化其位置嵌入；
3. 对于视觉形态，构建了多通道字符图像作为图形特征，每个通道对应一个特定的中文字体，使用ResNet对图像进行分块编码，得到字符图形标识；Pillow库提取汉字图像，当处理特殊记号(例如，BERT的[CLS]和[SEP])时，使用零值张量作为它们的图像输入；
4. 然后选择性融合多模态信息，预测在相应模态中给定输入的正确字符预训练语音和图形编码器；选择性模态融合模块有3个Transformer层，即L'= 3，预测矩阵与语义编码器的词嵌入矩阵相联系。所有嵌入和隐藏状态的维数都是768。

2.2 SpellGCN

SpellGCN (Cheng et al., 2020)是通过一个特殊的图神经网络将音似和形似的知识融合进语言模型，该模型构建了字符之间的一张图，SpellGCN通过学习将这张图映射到一组相互依赖的字符分类器上。然后，将这些分类器应用到从BERT中提取的文本表示上，并能够使整个网络进行端到端的训练。SpellGCN能够捕获发音和字形的相似性，并能够探索字符之间的先验依赖。尤其是，基于发音和字形之间的关联构造两张相似性图。SpellGCN将两张图作为输入，并在相似字符交互作用之后，为每个字符生成一个向量表示。然后，这些向量表示被构造成一个字符分类器用于BERT输出的语义表示上。

2.3 CRASpell

CRASpell算法(Liu et al., 2022)是一种上下文错误鲁棒的中文拼写纠错模型，现有的纠错模型存在两个问题：过纠正问题和多错误干扰。大部分纠错模型都基于BERT，而BERT本身是一个MASK语言模型，其预训练的方式导致倾向于预测高频表达，因此现有的纠错模型普遍存在对低频表达过纠正的问题。为了解决这个问题，CRASpell (Liu et al., 2022)在纠正网络中引入了Copy机制，增加模型对原字的预测概率，从而减少误纠正的情况。
在中文纠错中，多错误文本非常普遍（多错误文本指单个句子包含多个错别字）。纠错模型本质上是基于上下文对错别字进行识别和纠正，在多错误样本中，上下文中至少包含一个错别字，这种错别字使得上下文中包含噪声信息，相互干扰，导致模型在多错误文本上纠错效果差，针对这个问题，CRASpell (Liu et al., 2022)的基本思想：对于每个输入文本生成一个带噪样本；训练模型时，让纠错模型在带噪样本和原样本上输出的分布一致，提升模型对噪声的建模能力。
Correction Module的Transformer Encoder是一个12层的Base-base模型，Generative Block基于最后一层BERT的输出向量在整个字表空间进行生成。Copy Block计算从输入中进行Copy的概率，并基于生成概率和Copy概率得到最终的生成概率。如果生成的字和输入不同，则表示这里是一个错别字，生成的字即为纠正字，否则表示此处不是错别字。
Noise Modeling Module首先根据待纠错文本基于混淆集随机替换原样本中的字来构造噪声样本，然后基于噪声样本计算生成概率。训练模型时，让纠错模型在带噪样本和原样本上输出的分布相近，提升模型对上下文噪声的建模能力。噪声样本生成时只在错别字周边10个字符以内采样加噪位置，且70%的替换随机选择语音相似字符，15%的替换随机选择字形相似字符，15%的替换从词汇表中的随机选择。

2.4 MacBert4csc

MacBert4csc (Cui et al., 2020)是一个基于bert的中文文本纠错模型，在bert的网络基础上增加了一个全连接层作为错误检测（detection），利用detection层和correction层的loss加权得到最终的loss。其主要特征在于预训练时不同的MLM task设计：使用全词屏蔽(wwm, whole-word masking)以及N-gram屏蔽策略来选择candidate tokens进行屏蔽；BERT类模型通常使用[MASK]来屏蔽原词，而MacBERT使用第三方的同义词工具来为目标词生成近义词用于屏蔽原词，特别地，当原词没有近义词时，使用随机n-gram来屏蔽原词；和BERT类模型相似地，mask掉输入样本的15%，对于每个训练样本，80%的词被替换成近义词(原为[MASK])、10%的词替换为随机词，10%的词不变。

以上是对所有调研模型的介绍。

3 数据

比赛数据要求：赛道一允许使用任意开源数据用于训练。例如，可使用现有的真实开源数据集进行训练，如SIGHAN 2013、CLP 2014、SIGHAN 2015等，也可以使用伪数据Wang数据集。此外，SIGHAN 历年赛事中也给出了音近、形近混淆集（Confusion Set）作为参考，参赛者可按需使用。
本赛道提供基于YACLC-CSC 数据集的开发集与测试集。在拼写错误标注方面，YACLC-CSC继承前人的研究，规定只标注和修正“音近”和“形近”有关的错误。判定为“音近”或“形近”或“形音兼近”的依据来自相关的汉语语音学、文字学理论及对外汉语教学理论。标注过程采用多人标注再由专家审核的方式以保证标注质量。
我们在比赛过程中主要使用的是赛道一提供的以上数据集，包括SIGHAN6461、Wang共271451条数据，切分成训练集221451条、验证集50000条（因Lang8的错误类型和赛道一任务不同，没有使用）。测试集是赛道一评测数据集1100条。我们通过随机替换、音形相似替换、混淆字对替换几种方法按一定比例混合的方式替换label句子上的token，在训练集上分别进行字级别和词级别的数据增强。字级别数据增强即分析和搜集训练集和验证集badcase中误纠正和漏纠正的字对，针对训练数据进行字对混淆集的数据增强；词级别数据增强即对训练集和验证集badcase进行分词后统计获得误纠正和漏纠正的

词对，针对训练数据进行词对混淆集的数据增强。增强后训练集增加14万条。
混淆集举例：
字对：幺/么，的/得/地，着/著/薯，的/了，做/作，购/沟.....
词对：熔化/溶化/融化，嫉妒/忌妒，白薯/白著，.....

4 实验及结果

赛道一提供的是基于Bert的基线模型，基线中使用的预训练语言模型是bert-base-chinese。我们先后做了数据增强、模型微调、模型调研、对比实验等工作，以及引入混淆集、困惑度等策略，达到最终提交的结果，句子级别C-F：83.19。
下面我们将其中几个有明显提升效果的重要实验分别进行详细介绍。

4.1 实验一

我们先用赛道一提供的170条验证集做训练集，在基线模型上做微调，在赛道一测试集上进行测试，测试结果为49.34。

$$sentence\_level : \{'C-F' : 49.34,' D-F' : 55.43\}$$
$$character\_level : \{'C-F' : 85.44,' D-F' : 60.86\}$$

4.2 实验二

我们做了数据扩充，加入比赛提供的SIGHAN和Wang271k（因Lang8的错误类型和赛道一任务不同，没有使用）合成数据集27万，切分为训练集22万+验证集5万，再在基线模型上进行微调，赛道一测试集结果为：67.59。

$$sentence\_level : \{'C-F' : 67.59,' D-F' : 69.95\}$$
$$character\_level : \{'C-F' : 93.71,' D-F' : 74.37\}$$

4.3 实验三

我们又在22万训练集上做了数据增强，生成14万条增强数据，增强后的训练集为36万。我们用增强后的数据集在基线模型上进行微调。预测结果为73.4。

$$sentence\_level : \{'C-F' : 73.4,' D-F' : 75.69\}$$
$$character\_level : \{'C-F' : 96.32,' D-F' : 83.98\}$$

4.4 实验四

我们考虑到将传统CSC任务的Detection阶段单独拆出来做一个简单的判断句子中每个位置对错的二分类任务，以此来提高模型的检错性能，尽可能的把所有的错误位置都检查出来。
在纠错阶段我们结合N-gram计算困惑度进行纠错，同时设置了一些过滤规则，这样引入了一些额外知识保证纠错阶段的可控性，也使得纠错阶段的性能得以提高。我们将句子中出现错误的地方标记为1，其余位置都标记为0，在chinese-roberta-www-large预训练模型基础上结合SIGHAN + WANG增强后的数据集对检错任务进行微调。

4.4.1 数据预处理

将训练集中的标签数据和原始文本进行对比后，改变标签数据的标签，出现错误的位置改为1，其余位置改为0。训练阶段：将模型最终输出的结果映射到0、1分类空间，并以标签和预测结果的交叉熵作为损失函数。预测阶段：将所有识别到错误的位置替换为一个指定的汉字，只计算出来检错任务的指标。

4.4.2 实验细节

以roberta-large为预测模型，用SIGHAN+WANG的数据集进行微调，然后将roberta最后一层的输出映射到0、1二分类得到最终的预测结果。AdamW作为优化器，batch size=16，训

练了16 epoches，学习率=2e-5。
针对一些未识别出来的错误我们扩充了一下相应的错误数据。

4.4.3 实验结果

Model	dataset	C-D	S-D
01	**sigwang**	73.4	67.98
01	**enhanced**	89.16	85.43

Table 1: 实验一的结果

4.5 实验五

我们分别用ReaLiSe (Xu et al., 2021)、CRASpell (Liu et al., 2022)、Macbert4csc (Cui et al., 2020)模型，先后在增强前的22万和增强后的36万数据集上进行训练，在赛道一测试集上进行预测，得到不同的测试结果。

4.5.1 Macbert4csc模型

本次实验中，Macbert4csc模型以chinese-macbert-base作为预训练模型，batch size=32，训练了10 epoches，学习率=5e-5，得出的结果如Table 2 所示。其中sigwang是指使用SIGHAN+WANG271K数据集,enhanced是指使用增强数据集。

Model	dataset	f1 score	C-C	C-D	S-C	S-D
Macbert4csc	**sigwang**	69.47	81.45	65.86	69.47	70.40
Macbert4csc	**enhanced**	78.99	82.29	78.85	78.99	79.48

Table 2: 基于Macbert4csc的结果

4.5.2 CRASpell模型

CRASpell模型是以chinese-roberta-wwm-ext作为预训练模型，AdamW作为优化器，batch size=32，训练了15 epoches，学习率=3e-5。得出的结果如Table 3所示。

Model	dataset	f1 score	C-C	C-D	S-C	S-D
CRASpell	**sigwang**	52.52	81.19	65.11	52.52	61.09
CRASpell	**enhanced**	79.38	95.04	87.12	79.38	81.34

Table 3: 基于CRASpell的结果

4.5.3 ReaLiSe算法

ReaLiSe算法是以BERT-wwm作为预训练模型，AdamW作为优化器，batch size=32，训练了10 epoches，学习率=5e-5。得出的结果如Table 4所示。 中文拼写纠错，使用听觉和视觉信息有助于汉语拼写检查任务。构建的混淆集需要人工构建，混淆集预先定义和固定的，它不能覆盖所有的相似关系，也不能区分相似性中的差异。该算法利用了汉字的多模式信息来预测正确的输出。ReaLise (Xu et al., 2021)模型使用特定的语义、语音和图形编码器捕捉这些形式的信息，并提出一种选择性模态融合机制控制这些模态的信息流。SIGHAN基准显示，提出的算法比仅适用文本信息的基线模型具有更大优势，使用听觉和视觉信息有助于汉语拼写检查任务。我们采用RealiSe模型，得到的结果如Table 4所示。

4.5.4 实验结果

通过对比以上模型效果，每个模型的指标相差不大。但经过具体数据分析发现，不同模型对不同的错误改正各有优劣，具体表现如下：如ReaLiSe (Xu et al., 2021)在标点、字母等特殊符号以及易混淆词上表现较差；CRASpell (Liu et al., 2022)在音相近和某些助词上表现

Model	dataset	f1 score	C-C	C-D	S-C	S-D
ReaLiSe	**sigwang**	73.47	96.32	84.11	73.47	75.76
ReaLiSe	**enhanced**	80.77	96.89	88.06	80.77	82.31

Table 4: 基于ReaLiSe的结果

差；而macbert4csc (Cui et al., 2020)在常见的易混淆词上表现不错，但是在动词方面表现不佳。01模型在检错方面性能最佳。各个模型的对比如Table 5所示。

Model	dataset	f1 score	C-C	C-D	S-C	S-D
Macbert4csc	**sigwang**	69.47	81.45	65.86	69.47	70.40
	enhanced	78.99	82.29	78.85	78.99	79.48
CRASpell	**sigwang**	52.52	81.19	65.11	52.52	61.09
	enhanced	79.38	95.04	87.12	79.38	81.34
ReaLiSe	**sigwang**	73.47	96.32	84.11	73.47	75.76
	enhanced	80.77	96.89	88.06	80.77	82.31

Table 5: 各个实验的结果对比

4.6 实验六

基于以上实验结果，我们发现每个模型的纠错能力各有侧重，于是我们做出了新的构想，即把各个模型的检错结果进行合并，优势互补，尽可能全面地检测出句子中所有错误位置。我们把各个模型的纠错结果加入候选集，通过计算每个候选词的得分，从而得到最优结果。通过调研，我们决定采用N-gram计算替换每个候选词后句子的困惑度得分来决定选择最终结果。
通过分析，我们得出结论：N-gram纠错在垂直领域表现良好，因此，我们选用sighan数据作为N-gram语言模型的训练语料。另外，我们分别训练了字级别的5-gram模型和词级别的3-gram模型。在句子的困惑度的得分计算时，综合考虑字级别和词级别的得分，以此提高结果的可信度。

5 总结

在本次比赛中，我们采用了多模型结果融合的方法，结合解决过纠正问题和多错误影响问题的模型，使用汉字的多模态信息提升中文拼写纠错正确率的模型，及解决简繁转换问题的模型，在本次比赛的音形相近的问题上取得了较好的结果。

5.1 创新点

对于本次比赛，我们有以下创新点：1.多模型检错结果的融合技术：保持每个模型的特性，优势互补，尽可能全面地检测出句子中的错误位置。2.根据数据分析结果不断更新混淆集，并在此基础上进行数据增强，模型迭代。

5.2 不足

对于本次比赛，我们存在以下遗憾和不足：1.数据方面：数据量小。本次比赛主要使用数据仅限于sighan及wang271k数据，在进行数据增强处理时选取的方法单一。2.模型方面：本次比赛调研并使用的模型有限，未能涵盖目前所有的纠错模型，日后的工作中将更广泛地对比各个模型的优劣。未来我们也会尝试将目前热门的prompt范式使用在中文拼写检查任务中。

致谢

经过3个月的整理数据、模型调研、对比实验，我们最终完成了这次比赛。
首先，感谢我们的领导-方寸无忧公司CTO刘学谦给予我们这次参赛机会，及在我们比赛过程中给予的技术指导。感谢他一直以来对我们的支持和鼓励。从赛题的选择到最终完成，他都始终

给予我们耐心的指导和不懈的支持，他从他的专业角度耐心的指导我们模型的选型和数据增强的方法。
其次，感谢组织方，感谢智源平台(Wang et al., 2022)，感谢北京智源人工智能研究院提供给我们这次比赛机会，这是一次宝贵的历练，增强了我们团队的凝聚力，这将是我们的宝贵财富。
最后，衷心的感谢组织方的各工作委员会的老师们将在百忙之中参加我们的评测报告的评审工作。

参考文献

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online, July. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online, November. Association for Computational Linguistics.

Shulin Liu, Shengkang Song, Tianchi Yue, Tao Yang, Huihui Cai, TingHao Yu, and Shengli Sun. 2022. CRASpell: A contextual typo robust approach to improve Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3008–3018, Dublin, Ireland, May. Association for Computational Linguistics.

Yingying Wang, Cunliang Kong, Xin Liu, Xuezhi Fang, Yue Zhang, Nianning Liang, Tianshuo Zhou, Tianxin Liao, Liner Yang, Zhenghua Li, Gaoqi Rao, Zhenghao Liu, Chen Li, Erhong Yang, Min Zhang, and Maosong Sun. 2022. Overview of cltc 2022 shared task : Chinese learner text correction.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, listen, and see: Leveraging multimodal information helps Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 716–728, Online, August. Association for Computational Linguistics.