基于拼音编码和多轮纠错式推理方式的中文拼写检查纠错系统

周正茂	徐雪帆	陈志刚	韩伟	穆玉芝	张健
达观数据	达观数据	达观数据	达观数据	达观数据	达观数据
zhouzhengmao@	xuxuefan@	chenzhigang@	hanwei@	muyuzhi@	zhangjian@
datagrand.com	datagrand.com	datagrand.com	datagrand.com	datagrand.com	datagrand.com

摘要

中文拼写检查是中文自然语言处理领域中的热门任务，其目的是将错别字以一比一的方式正确修正。本文以此问题为基础，提出了一种串联式纠错系统。包括将拼音编码加入Transformer Encoder中，用多轮纠错的推理方法解决单句话多错误现象，使用困惑度计算将错误召回句子修正，使用实体纠错以及Ngram纠错方法补于系统尾部补充等。本文将从模型和数据方面详细阐述系统的原理和使用目的。实验表明，本文提出的系统可以显著提升纠错模型的性能。

关键词： 中文拼写检查 ; 自然语言处理

A Chinese Spell Check error correction system based on pinyin encoding and multi-round error-correcting inference

Zhengmao Zhou	Xuefan Xu	Zhigang Chen	Wei Han	Yuzhi Mu	Jian Zhang
Datagrand	Datagrand	Datagrand	Datagrand	Datagrand	Datagrand
zhouzhengmao	xuxuefan@	chenzhigang@	hanwei@	muyuzhi@	zhangjian@
@datagrand.com	datagrand.com	datagrand.com	datagrand.com	datagrand.com	datagrand.com

Abstract

Chinese Spelling Check is a popular task in the field of Chinese natural language processing, where the aim is to correct misspellings on a one-to-one basis. In this paper, a tandem error correction system is proposed based on this problem. This includes adding the pinyin encoding to the Transformer Encoder, using a multi-round error correction inference method to solve the phenomenon of multiple errors in a single sentence, using sentence fluent calculation to correct the incorrectly recalled sentences, and using Named Entity Recognition error correction as well as N-gram error correction methods to complement the system at the tail. This paper will elaborate on the rationale and purpose of using the system in terms of model and data. Experimental surfaces show that the system proposed in this paper can significantly improve the performance of the error correction model.

Keywords: Chinese Spelling Check , Natural Language Processing

1 引言

中文拼写检查（Chinese Spelling Check）任务一直是中文自然语言处理中一个比较有挑战性的任务。该文章主要介绍了一种新的纠错系统，平衡了训练和预测时任务的一致性，用审查句子困惑度，实体纠错的方法减少误召回。首先，在背景模块，我们介绍了中文拼写检查任务（Chinese Spelling Check）的相关情况，包括对该任务的描述和其难点；并且阐述了近年来NLP领域在中文拼写检查任务中作出的各种尝试，包括模型和数据集方面；我们还对实验数据进行了相关的数据分析。第二，在系统模块中，我们将详细介绍本文纠错系统使用的各种方法：1. 针对性地使用了以混淆集为基础的bert模型作为基础模型，并改编Transformer Encoder结构将拼音编码加入其中。2. 多轮纠错方法，通过每次只纠错概率最大的字来实现统一训练和预测任务；3. 通过检验句子的混淆度来减少“将正确句子纠成另一个正确句子”的情况。4. 通过实体抽取来减少模型对实体的错纠。5. 通过n-gram纠错模型进行错误的补充。第三，在实验模块中，我们详细阐述了该系统所使用到的训练数据，以及该方法在YACLC-CSC中进行的多次试验，并且通过消融实验验证了方法的有效性。最后，我们将对文章内容进行总结，并提出方法的局限性和未来可以提高的方向。该工作做出的独特贡献是：

1. 将拼音编码加入Transformer Encoder中强化模型对拼音的识别。
2. 并使用多轮纠错的方式统一训练和预测阶段的一致性。
3. 使用困惑度验证的方法减少纠错任务中的误召回。
4. 提出纠错系统，用多种方法串联提升效果。

本文所提及的工作代码开源于此[1]。

2 背景

2.1 任务情况

该工作参与的任务为CCL2022的中文拼写检查（Chinese Spelling Check）任务(Wang et al., 2022)，目的是检测并纠正中文文本中的拼写错误（Spelling Errors）。拼写任务包括：音近，形近，音形兼近。特点是错误内容与被替换内容长度相同，这也意味着输入语句与输出语句的长度相同。

输入句	捣好的年糕一定要干快吃。
输出句	捣好的年糕一定要赶快吃。

在上述表格描述的案例中，输入句中的“干”是一处音形兼近字的错误使用，应被替换为正确的“赶”，从而得到输出句子。

输入句	我做错了事情，真是湖图。
输出句	我做错了事情，真是糊涂。

除此之外，句子中也可能包含多处错误。如上表所示，输入句中的“湖图”两字都被错用，需要被正确改正为输出句子中的“糊涂”。

这种任务在实际应用中会遇到一些困难。1. out of vocabulary的问题会严重影响模型效果。首先，模型不可能纠错出没有在训练中见过的表达方式；其次，模型会由于在训练中没有见过相关实体而将他们错纠。2. 当一个句子存在多处错误，纠错时会受到上下文的影响，也就是受到其他错误的影响导致难以纠错成功。

比赛方主要提供了基于YACLC-CSC的数据集作为验证集和测试集。其中验证集共包含170条句子作为数据，85条句子存在错误，共有99个错字。这意味着存在单个句子包含多个错字。而测试集共有1100条。

[1]https://github.com/Gridnn/ccl22track1/tree/master/track1

错误类型	近形	近音	近音近形
数量	12	51	33

我们针对验证集的数据进行了相关数据分析，对85条句子和其包含的错误进行了人工分析，划分了它们的错误类型。我们可以看出绝大多数的错误是近音字或近音近形字，而纯粹的近形字非常少，只有12条，且其中9条都是“幺”和“么”的错误。所以，在后续混淆集过滤中，我们更针对于近音字进行处理；在模型方面，我们也会加强其对拼音的识别。而字符的形状和笔画将不会成为本工作的考虑对象。

2.2 相关工作

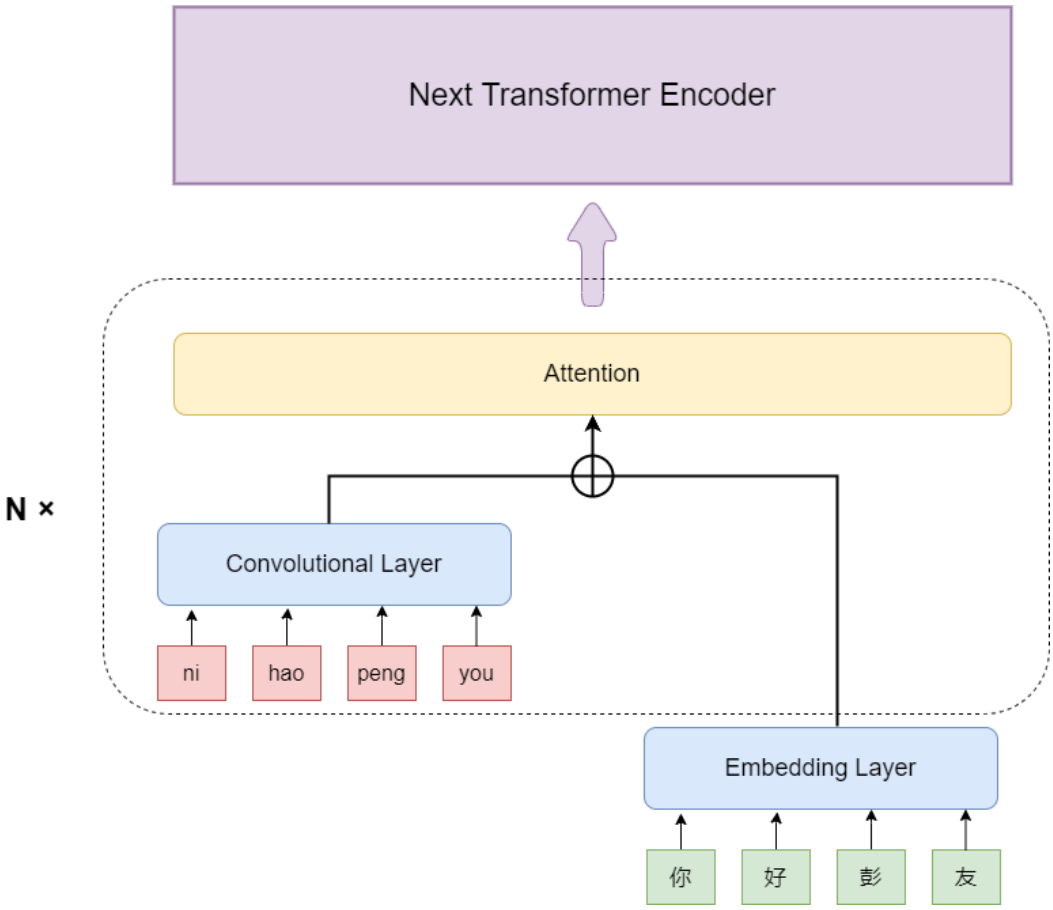
在中文拼写检查领域在近年来涌现了多个有价值的工作。在模型算法方面，DCN (Dynamic Connected Networks for Chinese Spelling Check) (Wang et al., 2021) 将单字的拼音转换为一个独特的整数进行编码加入模型中，使模型掌握拼音信息。并且在预测阶段通过拼音生成topk的备选单字，最终解码获得最优路径。PLOME (Pre-training with Misspelled Knowledge for Chinese Spelling Correction) (Liu et al., 2021) 将单个字的拼音和笔画信息全部进行编码并加入BERT的embedding层中。并且，有别于直接拿一个整数来代表一种拼音，该工作将所有的字母用token表示，然后用token表示拼音和笔画，最后经过一个GRU层来得到embedding。这样的好处是模型可以掌握拼音的构成信息，相似的拼音或笔画组合将获得相似的编码信息，从而加强模型对近音和近形字的识别。但由于GRU层的关系，该方法的效率较低。此外，该论文同时提出了cbert（confusion set based BERT），它是使用混淆集生成的训练数据来预训练的模型。模型结构和BERT一样。CRASpell (A Contextual Typo Robust Approach to Improve Chinese Spelling Correction) (Liu et al., 2022) 通过自动在错字周围生成其他错误来解决模型难以处理一句话存在多个错误的问题。该方法通过生成多个错误以达到在训练时模拟“单句多错”的情况，最终在单句多错的测试集中达到了好的效果。

而在数据集方面，SIGHAN15 (Tseng et al., 2015)收集了在台湾进行的基于计算机的对外汉语考试（TOCFL）的作文部分作为语料，并让汉语母语者进行人工标注，得到了精准度很高并且贴近实际错误的高精度数据集。而人工标注的问题是无法获取巨量数据，为了解决人工标注数据较少的问题，A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check (DingminWang et al., 2018) 通过采集OCR和ASR的数据获取近音和近形字数据集，通过使用该混淆集，可以自动在任何数据下生成大量的错误-正确的句子对。

3 系统

上述流程图简单表述了我们系统的纠错流程。主要由五个部分构成。首先，我们对基础模型进行预训练和微调，然后进行多轮纠错式推理，第三步是使用训练好的困惑度模型进行误召回检查，第四步则是使用实体纠错方法对于相关实体再次审核，最后使用精度很高但召回较低的ngram模型进行再一次的补充。详细内容的介绍将会在该模块下逐一展开。

3.1 拼音编码基础模型



在本次工作中，我们使用了DCN (Wang et al., 2021)中所采用的拼音编码方法, 即将完整的一个拼音组合编码成一个整数，输入模型中。而后，将拼音经过Convolutional layer后的矩阵与文本经过Eembeding layer之后的矩阵相加，再输入到Attention中，以此强化拼音编码在模型中的权重。不同于DCN中只在Embedding阶段将拼音编码与字符编码相组合，我们借鉴了DeBERTa (He et al., 2020)中所提及的Disentangled attention方法，此过程将在每一次的Transformer encoder中进行重复。其公式表示如下：

$$c_i = Conv(p_{i-1}, p_i, ..., p_{i+1})$$

$$o_i = c_i + h_i$$

其中c_i是拼音编码，而h_i则是字符的编码。这种拼音编码的形式比较简单，没有声调也忽略了多音字。但其运算效率高，在实际测试中与复杂的编码方式相比的效果也相近。而我们没有采取任务字形方面的编码主要是由于在对验证集的统计中，我们发现绝大多数的错误都是音近或音近形近导致，同时，统计结果也表明中文纠错中80%以上的错误都是音近字导致 (Liu et al., 2010)。所以我们将重心集中于拼音之上。

我们借鉴cbert (Liu et al., 2021)中文拼写检查模型，在大规模语料中进行训练和微调，对于无标注数据则采用混淆集自动生成数据的方法进行构造。cbert是基于混淆集构造的bert模型，由于我们需要大量未标注数据来扩大模型训练的数据量，使用混淆集来生成纠错错误对变得尤为关键。而cbert所表述的方法与我们的要求一致，所以我们采用了该方法。

3.1.1 参数介绍

除拼音编码的部分外，模型其余构造和base bert (Devlin et al., 2019)的结构相同。共有12个transformer层，hidden units的大小为768，attention head共有12个。而输出层则是拼接了一个输入维度为768，输出维度为单词数量的全连接层。最终再使用Softmax计算出每个备选单词的概率。

3.1.2 Loss function

我们采用交叉熵作为模型训练和微调的损失函数。该方法是多分类问题中常见的损失函数，具体公式如下：

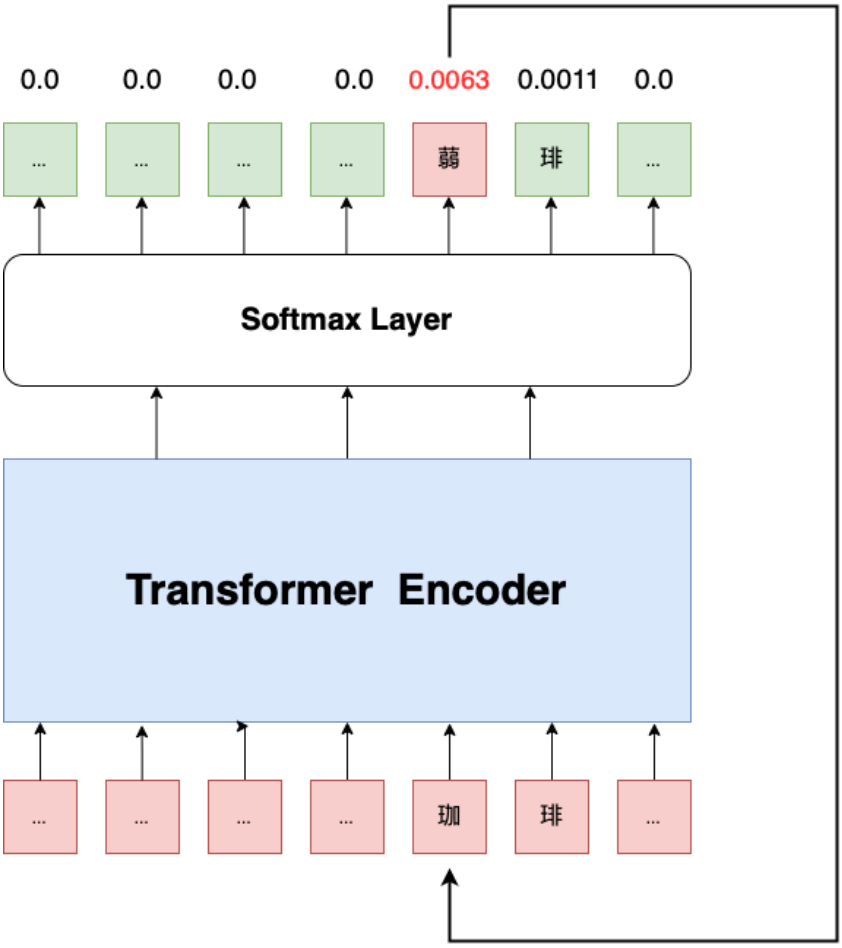
$$L = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic} log(p_{ic})$$

其中，M是备选字的总数，y_{ic}取值为0或1，在样本i中等于备选字则为1，其余为0。p_{ic}是样本属于每个备选字的概率。

3.1.3 Optimization

我们采用AdamW (Loshchilov and Hutter, 2017)作为模型训练和微调的优化器。AdamW为了解决自适应学习率的问题提出了权重衰减，相比于Adam (Kingma and Ba, 2014)加L2正则化来说有更高的计算效率。

3.2 多轮纠错方法



在中文拼写检查领域，解决同一句话存在多个错误的问题一直是一项重要的挑战。其原因是存在错误的上下文会对预测产生影响。在之前的工作中，CRASpell (Liu et al., 2022)通过在训练时随机在错字周围生成新的错字来模拟单句多错字的环境，但这种方法也会对数据的真实分布产生影响。一般的Transformer类纠错模型，会在每个位置返回其概率最大的字，如果存在多个错字，也将一并返回。而我们对此提出了多轮纠错的方法。如上图所示，具体来说，如果

一句话中存在多个错误，在模型预测阶段，我们每次只选取预测错误概率最高的字，将其更正后放回原句，再进行第二轮的纠错，直到不再出现新的错误的句子。

轮数	句子	概率
输入句	我最近每天在珈琲馆念七个小时以上的书。	
第一轮	我最近每天在蒻琲馆念七个小时以上的书。	0.0063
第二轮	我最近每天在咖琲馆念七个小时以上的书。	0.1605
第三轮	我最近每天在咖啡馆念七个小时以上的书。	0.9978
第四轮	我最近每天在咖啡馆念七个小时以上的书。	无变化

以上方表格为例，第一轮纠错模型将“珈”纠错成“草头弱”字，在第二轮再把“草头弱”纠错成“咖”，最后由于“咖”对于上下文的补充，本来无法纠出的“王非”也被以非常高的概率成功纠错成“啡”。该方法将所有单句多错字问题转换成了单句单错字问题，使得训练与预测的任务更具有一致性，分布更加统一，并且没有丢弃任何输入信息。

3.3 困惑度减少误召回

误召回问题同样是中文拼写检查中面临的挑战。由于纠错任务本质是选取位置上概率最高的字，所以会不时出现”用正确的字替换正确的字“的情况。我们通过对比句子修改前后的困惑度来减少误召回的情况 (Bao, 2021)。该方法主要由以下几个步骤组成:

1. 针对一句话，将每个字依次[MASK]。例如：我在吃饭。将被MASK成四个sequence：[MASK]在吃饭，我[MASK]吃饭，我在[MASK]饭，我在吃[MASK]。
2. 将上述四个句子经过tokenizer编码后输入进模型。
3. 返回所有[MASK]字符对于原字的预测概率。记为$P = [p_1, p_2, p_3, ..., p_n]$
4. $l_{score} = sum(log_2(p_i))$, where i = 1, 2, 3, ..., n
5. $score = 2^{-l_{score}}$

故此，socre较大的句子较socre较小的句子更为不通顺。

输入/输出句	困惑度
我知道他很忙，没有时间跟我联系。	1.2539
我知道他很忙，没有时间跟我练习。	1.6251

上述表格中的例子显示，模型将“我知道他很忙，没有时间跟我联系。”误纠成“我知道他很忙，没有时间跟我练习。”，由于输入句本身就是通顺的句子，所以困惑度并未因此上升，这个误召回的情况也就可以被成功减少。该方法对于纠错的整体性有着针对性的提高，会减少出现句子部分纠错正确但未全部纠错正确的情况，弊端是对于字符级别的纠错性能可能带来损失。

3.4 实体纠错

当错字出现在诸如人名、地名、作品名的实体上时往往常规的纠错模型方案很难达到优异的表现，因为纠错模型的学习任务里不包含辨识实体的能力，而语料也不能覆盖所有可能的实体名，所以不免会出现在实体上的错纠或漏纠。

输入	输出	问题类型	实体类型
因为我想看绳文杉。	因为我想看绳文彬。	错纠	地名
《格得战记》也有汉语配音和字幕。	《格得战记》也有汉语配音和字幕。	漏纠	作品名

以上方表格为例，第一条样例是把日本的地名[绳文杉]错纠成[绳文彬]，这是因为模型不具备绳文杉是个地名的知识，而[文彬]的字词组合在模型输出中概率更高；第二条样例是漏纠

了[格得战记]的错误，正确的作品名是[格德战记]，这是因为模型学习语料中没有覆盖这个作品名称。为了解决实体误纠、漏纠的问题，我们训练了命名实体识别模型，模型结构是以Bert base (Devlin et al., 2019)加CRF的序列标注模型。预测时将实体词典和命名实体识别模型进行一定策略排布，具体来说：

1. 当输入文本命中实体词典，则命中部分保留原文，不采纳纠错模型的纠错结果；
2. 将文本输入命名实体识别模型，得到实体预测结果。对于不在实体词典中的实体，假设实体可表示为$Entity = C_1C_2C_3...C_n$，C_i为实体的第i个字。同时令$ConfusionSet_{C_i}$为C_i的同音形近混淆集，集合的容量为$Size_{C_i}$，则对于实体Entity共有$\prod_{i=1}^{n} Size_{C_i}$种同音形近排列枚举值，对于每一个枚举值都进行实体词典匹配，选择匹配命中且与原文相比改动最小的实体词作为纠错结果。如果没有一个枚举值能够命中实体词典，则实体部分保留原文，且不采纳纠错模型的纠错结果。

3.5 Ngram纠错方法

Ngram方法采用无监督方式纠错，即通过无标注语料训练n元语言模型，使用n元语言模型进行错误检测与纠正，通常认为ngram纠错方式在准确率方面效果一般，但结合过滤策略可在一定程度上提升准确率，即增强语义信息方面的检测，解其余方式带来的误纠。n元语言模型，一个语言模型构建字符串的概率分布$P(W)$，假设$P(W)$是字符串作为句子的概率，则概率由下边的公式计算：

$$P(W_1W_2...W_n) = \prod_i P(W_i|W_1W_2....W_{i-1})$$

其中W_i表示句中第i个词。$P(W_4|W_1W_2W_3)$表示前面三个词是$W_1W_2W_3$的情况下第四个词是w_4的概率。$W_1W_2...Wi-1$称作历史，如果w共有5000个不同的词，$i=3$的时候就有1250亿个组合，但是训练数据或已有语料库数据不可能有这么多组合，并且绝大多数的组合不会出现，所以可以将$W_1W_2...W_{i-1}$根据规则映射到等价类，最简单的方式就是取W_i之前$n-1$个历史，根据马尔科夫假设，一个词只和他前面$n-1$个词相关性最高，这就是n元语言模型。

$$P(W_1W_2...W_n) \approx \prod_i P(W_i|W_1W_2....W_{i-1})$$

我们在无监督数据集中训练出该模型，并用比较高的阈值严格过滤结果，起到少召回的效果，在纠错系统的最终步骤完成对结果的补足。

3.6 数据集生成

由于先前提到的out of vocabulary对于纠错任务的严重影响，我们需要模型在尽可能大的数据集中进行训练。由此，需要使用混淆集在无标注数据下自动生成训练集。所有用到的混淆集和数据生成的代码均在项目文件中开源。我们使用的混淆集具体说明如下。

3.6.1 数据生成

针对未标注数据集，我们采用如下方法进行自动纠错对的生成。

1. 使用分词算法将原句子进行分词处理。
2. 使用序列标注模型对所有词语进行属性标注、
3. 对被序列标注模型所标注出的人名，地名类词语（如xx酒店，xx公司）进行不设错处理，即不会被替换为错字。同样被过滤的还有非中文词和停用词。
4. 随机按比例抽取字词进行改动。如该词在混淆集中，15%不改动，15%概率随机改动，70%概率在混淆集中随机抽取改动。如该词不在混淆集中则不改动。

3.6.2 混淆集

关于近音字，在使用混淆集进行过滤任务时，我们采用以下算法进行近音字判定：考虑所有的多音字情况，当A字与B字所含拼音字母的差异小于2个，即判定为近音字。在使用混淆集进行生成任务时，则采用储备的混淆集直接进行替换。而关于近形字，由于先前对验证集数据的分析，我们认为绝大多数的错误都是音近或音近形近的错误，所以在使用混淆集进行生成任务时，我们没有使用形近字混淆集。而在过滤任务时则采用储备的形近字混淆集进行过滤。

4 实验

4.1 预训练

4.1.1 数据

我们使用了混淆集过滤后的，从维基百科中抽取的1-1的纠错语料共425568条。在微信和新闻语料的基础上，使用混淆集生成的纠错语料共160万条。总共得到超过200万的纠错句子对。

4.1.2 参数设置

参考原论文 (Liu et al., 2021)，我们将学习率设置成5e-5，将batch size设置成64，将最大句子长度设置成128。

4.2 微调

4.2.1 数据

在微调阶段，我们使用了SIGHAN15 (Tseng et al., 2015)的训练数据5167条，YACLC-CSC的验证数据170条，WANG27 (DingminWang et al., 2018)训练数据271280条，Hybrid训练数据264946条。这些训练数据都已经是句子对的形式，所以没有使用混淆集。

4.2.2 参数设置

在微调阶段，我们将学习率设置成[1e-4, 2e-5, 5e-5]，将batch size设置成[16, 32, 64]，将最大句子长度设置成128。

4.3 实验结果

依照比赛的要求，我们使用F1作为验证模型效果的指标，其中包括纠错F1和检测F1，并且分为句子和字词两个级别，其数据将以省略百分号（*100）的形式呈现。以下表格为模型在YACLC-CSC的测试集上的表现。值得注意的是，字词级别的Correlation F1只检查被模型检测到的错误，而不是所有错误。举例来说，假设模型只修改了一处错误，且修改正确，字词级别的Correlation F1将是100。

	Sentence Level		**Character Level**	
	Correlation F1	Detection F1	Correlation F1	Detection F1
Baseline	50.05	56.52	84.67	61.94
PYbert	79.88	81.83	97.14	85.95
PYbert + Multi-round	80.27	82.02	97.04	86.0
PYbert + Multi-round + Fluent	81.55	82.23	97.3	86.21
PYbert + Multi-round + Fluent+ NER	83.24	84.6	97.74	87.81
PYbert + Multi-round + Fluent+ NER + Ngram	84.33	85.49	98.32	88.34

在如上表格中，Baseline是官方提供的基线模型；PYbert为加入拼音编码的基础模型；Multi-round为多轮纠错方法；Fluent为困惑度检测方法；NER为实体纠错方法；Ngram为Ngram纠错方法。

在上述所有方法中，多轮纠错方法会增加召回数量，找到部分原先无法纠错的案例。困惑度检测则会减少召回的数量，将部分错误纠正的句子删除，但同时也会损失少量正确的句子。实体纠错方法则是针对所有实体进行再一次审查，将未正确纠错的实体正确纠错，将错误纠错的实体删除，准确率较高。最后，Ngram方法将补充未找到的案例，其召回数量很低（在该1100个句子的案例中只召回88条），所以不会删除错误召回。此方法的使用顺序大体基于模型的准确率（precision）。准确率较高而召回率（recall）较低的方法将放置于系统尾部，保证其输出的结果被删改的概率更低。如图所示，基础的拼音模型在经过生成数据的预训练和其余的微调过后，效果对比基线模型有大幅提升。而后，系统的每一个步骤都对Correlation F1的提升起到了帮助。

5 总结

文本提出了一种针对于中文拼写检查任务的纠错系统，并对其主要包含的五个部分进行了详细描述。并给出了该系统所进行的实验。在模型方面，我们提出了将拼音编码进Transformer Encoder的模型结果；针对于单句多错情况的多轮纠错方法；更加考虑句子整体性的混淆度检查方法；针对实体错误的实体纠错方法等。在数据方面，我们针对多音字进行数据增强，通过混淆集在巨量语料中自动生成正确-错误句子对。我们对于该系统进行了相关实验，验证了每个部分的有效性。但本工作同样存在如下缺点：1. 纠错系统较重，其中多个部分需要进行模型训练，便捷性不足。2. 拼音编码采用的将所有拼音组合编码为一个整数的方式无法识别多音字。3. 困惑度检测时可能将部分纠错正确但未完全纠错正确的句子判定为误召回。

参考文献

Bao, J. (2021). nlp-fluency. https://github.com/baojunshan/nlp-fluency.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

DingminWang, Song, Y., Li, J., Han, J., and Zhang, H. (2018). A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium.

He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Liu, C.-L., Lai, M.-H., Chuang, Y.-H., and Lee, C.-Y. (2010). Visually and phonologically similar characters in incorrect simplified Chinese words. In *Coling 2010: Posters*, pages 739–747, Beijing, China. Coling 2010 Organizing Committee.

Liu, S., Song, S., Yue, T., Yang, T., Cai, H., Yu, T., and Sun, S. (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. Association for Computational Linguistics.

Liu, S., Yang, T., Yue, T., Zhang, F., and Wang, D. (2021). PLOME: Pre-training with misspelled knowledge for Chinese spelling correction. In *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)*, pages 2991–3000, Online. Association for Computational Linguistics.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization.

Tseng, Y.-H., Lee, L.-H., Chang, L.-P., and Chen, H.-H. (2015). Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37, Beijing, China. Association for Computational Linguistics.

Wang, B., Che, W., Wu, D., Wang, S., Hu, G., and Liu, T. (2021). Dynamic connected networks for Chinese spelling check. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2437–2446, Online. Association for Computational Linguistics.

Wang, Y., Kong, C., Liu, X., Fang, X., Zhang, Y., Liang, N., Zhou, T., Liao, T., Yang, L., Li, Z., Rao, G., Liu, Z., Li, C., Yang, E., Zhang, M., and Sun, M. (2022). Overview of cltc 2022 shared task : Chinese learner text correction.