CCL23-Eval 任务6系统报告：基于预训练语言模型的双策略分类优化算法

黄永清[1]，杨海龙[1]，傅薛林[2]
[1]广东工业大学/ 广东省广州市
[2]桂林理工大学/ 广西省桂林市
1486590231@qq.com
hlyanggdut@aliyun.com
1735573894@qq.com

摘要

诈骗案件分类问题是打击电信网络诈骗犯罪过程中的关键一环，根据不同的诈骗方式、手法等将其分类，通过对不同案件进行有效分类能够便于统计现状，有助于公安部门掌握当前电信网络诈骗案件的分布特点，进而能够对不同类别的诈骗案件作出针对性的预防、监管、制止、侦查等措施。诈骗案件分类属于自然语言处理领域的文本分类任务，传统的基于LSTM和CNN等分类模型能在起到一定的效果，但是由于它们模型结构的参数量的限制，难以达到较为理想的效果。本文基于预训练语言模型Nezha，结合对抗扰动和指数移动平均策略，有助于电信网络诈骗案件分类任务取得更好效果，充分利用电信网络诈骗案件的数据。我们队伍未采用多模型融合的方法，并最终在此次评测任务中排名第三，评测指标分数为0.8625。

关键词： 预训练语言模型；深度学习；文本分类；诈骗案件分类；Nezha

System Report for CCL23-Eval Task 6: Double-strategy classification optimization algorithm based on pre-training language model

Yongqing Huang[1], Hailong Yang[1], Xuelin Fu[2]
[1]Guangdong University of Technology / Guangzhou City , Guangdong Province
[2]Guilin University of Technology / Guilin City , Guangxi Province
1486590231@qq.com
hlyanggdut@aliyun.com
1735573894@qq.com

Abstract

The classification of fraud cases is a key link in the process of cracking down on telecom network fraud crimes. According to different fraud methods and techniques, it will be classified. Through effective classification of different cases, it can facilitate statistics and help public security departments to grasp the distribution characteristics of current telecom network fraud cases. Then it can make targeted prevention, supervision, stop, investigation and other measures for different categories of fraud cases. The classification of fraud cases belongs to the text classification task in the field of natural language processing. The traditional classification models based on LSTM and CNN can play a certain effect, but it is difficult to achieve the ideal effect due to the limitation of the number of parameters in their model structure. In this paper, based on the pre-training language model Nezha, combined with anti-disturbance and exponential moving average strategies, it is helpful to achieve better results in the classification task of telecom network fraud cases and make full use of the data of telecom network fraud

第二十二届中国计算语言学大会论文集，第184页-第192页，哈尔滨，中国，2023年8月3日至5日。
(c) 2023 中国中文信息学会计算语言学专业委员会

cases.Our team did not adopt the method of multi-model fusion, and finally ranked third in this evaluation task, with the evaluation index score of 0.8625.

Keywords: Pre-trained language models , Deep learning , Text classification , Classification of fraud cases , Nezha

1 引言

电信网络诈骗利用电信网络技术手段实施诈骗，随着互联网技术的快速发展，逐渐演变出多种诈骗方式，诈骗严重危害和侵犯了公民的财产权，同时也对社会秩序和社会治安造成一定程度的损害。对现有的诈骗案件进行分类是打击电信网络诈骗犯罪过程中的关键一环，根据不同的诈骗方式将其分类，能够便于统计诈骗案件，有助于公安部门掌握当前电信网络诈骗案件的分布特点，进而能够对不同类别的诈骗案件作出针对性的预防、监管、制止、侦查等措施，面向电信网络诈骗领域的案件分类对智能化案件分析具有重要意义。诈骗案件分类属于自然语言处理(Natural Language Processing)领域的文本分类任务，是一项基础且重要的任务，其目标是将给定的句子或段落等文本通过算法模型归类为某个具体的标签，这需要模型能够从给定的文本信息中学习到句子的语义特征信息，从而才能够对文本进行准确的识别和分类。深度学习是由机器学习发展而来，并且深度学习在自然语言处理领域得到了大量的应用，包括但不限于文本分类、机器阅读理解、机器翻译等任务。

2 相关工作

文本分类算法主要分为基于传统机器学习的文本分类算法和基于深度学习的文本分类算法。基于深度学习的文本分类算法是近年来的研究热点，本节将分析文本分类方向基于LSTM、CNN和预训练模型的研究进展。

2.1 基于LSTM的文本分类算法

循环神经网络RNN利用序列的时序化特征信息对数据进行建模，并且在自然语言处理领域得到广泛应用。但是RNN网络由于其自身结构的缺陷，在网络的训练过程中会出现“梯度消失”或“梯度爆炸”问题，导致模型不能很好地拟合数据。于是RNN的改进版本LSTM模型被提出用于文本分类任务，能很好地解决文本之间长距离的依赖问题。张云翔等人采用长短期记忆网络进行文本分类任务，通过门控机制对输入信息进行选择性长期记忆 (张云翔, 饶竹一, 2020)。LSTM模型对单向的语义信息建模，没用考虑到反向内容的语义信息，有技术人员在电网领域设备故障文本的分类任务中，提出Attention-BiLSTM (田园, 马文, 2020)算法模型，采用双向LSTM网络模型提取文本的上下文信息，并融合注意力机制来捕捉文本的关键信息，从而提高文本分类的效果。研究人员提出层次文本分类任务和LSTM集合的联合嵌入方法 (Zhao and Ma, 2020)，充分利用了上层和下层标签之间的联系。GRU网络模型也是RNN模型的一种改进变体，模型拥有两种门控机制：更新门和重置门，相比于LSTM模型拥有更快的收敛速度，于是学者提出了一种基于混合注意力的GRU模型hatt-GRU (Wang et al., 2019)用于多标签的投诉文本分类，以数据中的字符构造文本向量，然后提出了一种混合注意力机制，通过分析角色跟情感特征的相关性来筛选出对分类贡献更大的特征，从而提高模型分类的精度。

2.2 基于CNN的文本分类算法

卷积神经网络最初应用于计算机视觉领域，在图像分类、目标检测等领域取得不错的成绩。Kim(2014)第一次将卷积神经网络应用在文本分类任务中，提出了一种网络模型名为TextCNN，利用多个不同大小的卷积核提取输入句子中的核心信息，之后根据最大池化操作选择出最具有代表意义的高维分类特征，接着再经过全连接层提取文本深度特征后根据softmax函数进行分类。但是因为TextCNN网络中卷积核尺寸通常不会很大，会导致面对长文本时无法有效提取长距离特征，2017年，由腾讯AILab提出的DPCNN (Johnson and Zhang, 2017)网络可以通过加深网络，能够抽取长距离的文本依赖关系，在深层网络中添加了残差连接，能够有效减缓梯度弥散问题。研究人员为了充分利用CNN和RNN各自的优点，对两者进行组合搭建分类模型，学者Lai et al. (2015)使用循环神经网络来建模文本的上下文语义信息，然后通过最大池化去提取关键特征信息用于分类，可以进一步提高分类的精度。

2.3 基于预训练语言模型的文本分类算法

自Transformer (Vaswani et al., 2017)模型发布以来，便揭开了预训练语言模型的序幕。谷歌提出的BERT (Devlin et al., 2018)是由Transformer中Encoder组成的双向自编码预训练语言模型，相较于过去的RNN、CNN等模型，BERT可以同时利用上下文信息进行训练，并且能够解决远距离依赖问题。在进行下游任务过程中，使用预训练好的BERT模型已经能够提取到丰富的句子特征，再通过[CLS]这个token的信息输入到全连接层就可以实现分类任务，并且当时在大部分数据集上取得了较好的结果。为了利用bert模型强大的编码能力，Lehečka et al. (2020)将bert模型运用到多标签文本分类任务中，并且在此基础上添加了池化层结构，将最后一层[CLS]的向量结合池化序列输出提高最后的分类精度。还有研究人员会结合BERT和RNN系列模型用于分类任务，作者在中文短文本分类任务上 (郝婷, 王薇, 2023)利用bert模型编码文本词向量，然后通过bilstm网络去提取上下文的语义特征，所提出的方法在评价指标上有良好的效果。除了序列模型可以结合bert模型，卷积神经网络结合bert模型同样可以加强语义表达的能力，张小为等人在新闻文本分类方面，结合bert与cnn模型，其准确率比原BERT模型的准确率多了0.31%,且更为稳定 (张小为, 邵剑飞, 2021)。随着bert模型的发布，有许多基于bert的改进模型，如ernie (Sun et al., 2019)、roberta (Liu et al., 2019)、albert (Lan et al., 2019)以及基于中文的bert-chinese-wwm (Cui et al., 2021)等模型，都可以在之前的学者研究中替换对应的bert模型完成对应的下游任务。

3 算法模型设计

3.1 Nezha预训练语言模型

Nezha模型 (Wei et al., 2019)是华为开源的一款基于中文的预训练语言模型。模型结构基于BERT模型，并且在其基础上做了一些改进和优化，主要改进是使用相对位置编码以及使用whole word masking(Cui et al., 2021)策略。在Transformer模型中对于文本的位置编码使用的是正余弦函数编码，在BERT模型中使用的是参数式位置编码，这两者都是使用绝对位置编码用于表示输入序列中每个字符的绝对位置信息，但是绝对位置编码的受到长度限制，无法处理超过预先设定长度的序列，所以BERT模型规定输入的文本长度最大不能超过512，并且绝对位置编码没用考虑字符之间的相对重要性。Nezha模型在计算自注意力时采用函数式相对位置编码，计算公式如下所示：

$$a_{ij}[2k] = \sin\left(\frac{j-i}{10000^{\frac{2\cdot k}{d}}}\right) \tag{1}$$

$$a_{ij}[2k+1] = \cos\left(\frac{j-i}{10000^{\frac{2\cdot k}{d}}}\right) \tag{2}$$

公式中i和j表示两个位置信息，通过两者相减引入相对位置信息，d表示词向量的维度大小，k是位置编码向量中的某一维度，根据位置索引的奇偶性分别用余弦或正弦的方式计算具体数值，最后得到长度为d的表示位置信息的向量。

3.2 分类模型结构

本文使用的模型结构如图1所示，实验采用nezha-base模型，总共有12层编码层，预训练语言模型每一层学习到的信息特征是不一样的，高层网络学习到的是高级语义信息，而低层则是较为普通的语言学特征 (Jawahar et al., 2019)。本次实验过程中通过对12层的编码向量与embedding的cls动态加权平均，加权系数是可学习参数，初始化时赋予最后一层encoder的cls向量较大值，然后让模型在训练过程中通过参数学习进而改变加权系数达到动态加权的目的，通过结合模型各层的表征向量可以达到增强向量语义的目的。

3.2.1 输入层

在输入层中，输入向量由词向量(word embedding)和段向量(segment embedding)相叠加而成，[CLS]和[SEP]符号分别用作一句话开头和结束的标记，并且[CLS]向量可以表示整句话的语义，且通常被用作下游的分类任务。在BERT模型中，输入层由三个向量组成，除词向量和段向量外，还有一个用于表示输入序列位置信息的位置向量(position embedding)，可以获取词与词之间的位置关系。在BERT模型中，位置向量与词嵌入编码类似，通过随机初始化一个位置

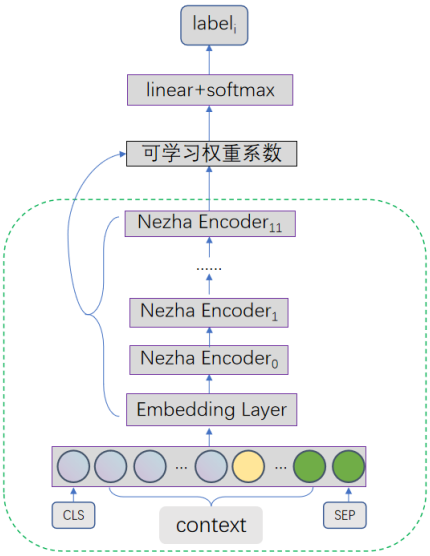

Figure 1: 基于Nezha动态加权的分类模型结构图

向量，然后在训练过程中对其优化，而transformer模型的位置编码由三角函数计算得到，为每个不同位置的单词生成一个位置向量，计算公式如下：

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{3}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{4}$$

式子中pos表示位置，i表示对应的维度，d_{model}表示词向量的维度，在transformer模型中取值512，根据维度位置的奇偶性，通过余弦或正弦的形式表示其位置信息。

3.2.2 编码层

编码层是整个模型结构的核心部分，通过base模型利用12层双向编码器对文本进行语义提取。每一层的编码器都包含多头注意力(multi-head attention)以及前馈神经网络(feed forward neural network)两部分，前馈神经网络思想比较简单，包含两个全连接层，通过矩阵维度的变换得到最后的输出矩阵，增强向量的表达能力。

多头注意力由自注意力机制组成，计算过程如图2所示，在计算过程中初始化h组分别计算各自的自注意力值，通过并行的方式计算完成h组之后再拼接并输入到一个线性层中形成完整的多头注意力。图中的X是输入矩阵，每一行代表一个词，长度表示维度大小，经过自注意力加权后，每个词都包含文本中其他所有词的信息，词与词之间的权重系数越大，则表明它们之间的相关性也越大。

在计算自注意力机制过程中，查询向量Q、键向量K和值向量V，都是由同一个源向量通过三个线性层进行变换得到，然后通过计算注意力分数来学习词与词之间的语义关系，首先利用查询向量Q与键向量K计算相似权重系数QK^T，再利用softmax函数对得到的权重系数进行归一化，让权重系数所有元素之和相加为1，最后将权重向量与值向量V相乘便得到了注意力矩阵A，计算过程如下：

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d}})V \tag{5}$$

上式表示单个自注意力机制的计算过程，只需要重复计算h次，然后把输出合并起来便得到多头注意力值。以transformer中多头个数8为例，在实际计算过程中，是通过矩阵的维度变换实现多头注意力计算，可以加快运算速度，计算公式为：

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V, i = 1,2,...,8 \tag{6}$$

第二十二届中国计算语言学大会论文集，第184页-第192页，哈尔滨，中国，2023年8月3日至5日。

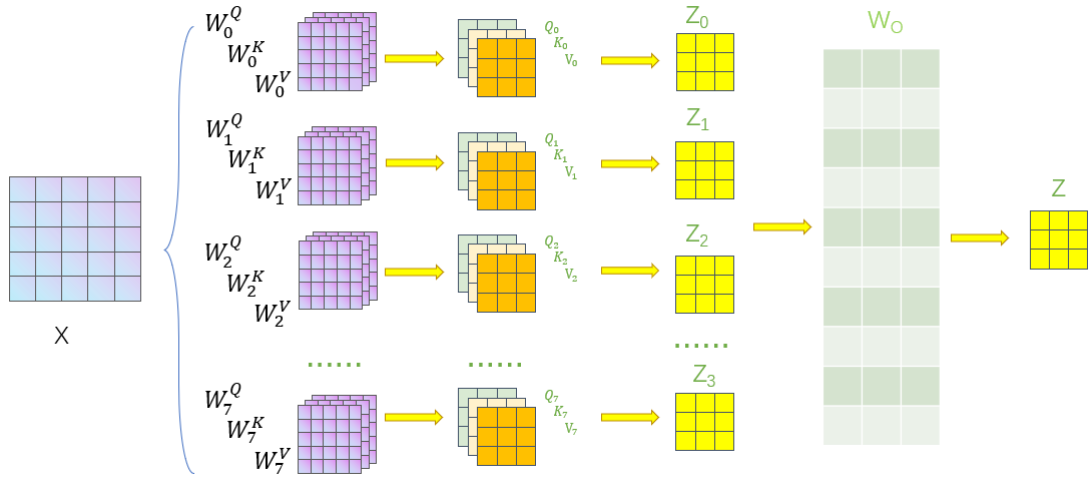

Figure 2: 多头注意力机制计算过程

$$head_i = Attention(Q_i, K_i, V_i), i = 1, 2, ..., 8 \tag{7}$$

$$Multihead(Q, K, V) = Concat(head_1, ..., head_8)W^O \tag{8}$$

3.2.3 输出层

输出层主要是用来预测文本对应的标签，最开始输入的文本序列$X = [X_1, X_2, ..., X_n]$经过输入层和编码层之后，得到的是融合上下文的动态词向量矩阵$E = [E_1, E_2, ..., E_n]$，然后通过全连接层将维度变换到与类别数量相同，最后结合softmax函数预测概率最大的类别作为输出。

3.3 对抗扰动策略

随着深度学习在各个领域蓬勃发展，关于对抗样本的研究也受到越来越多的关注。在计算机视觉领域，可通过对深度学习模型进行对抗攻击或者防御来提高模型的鲁棒性。在自然语言处理领域，对抗训练更多是作为一种正则化的方法来提高模型的泛化性能。在计算机视觉领域中，图像可以看作是一个连续实数向量，因此可以很容易加上一个很小的实数向量作为扰动，从而形成一个对抗样本，但在自然语言处理领域中输入的是一段文本，本质上是one-hot向量，因此不存在所谓的小扰动。在NLP中添加对抗扰动策略通常是对词向量矩阵W_E进行对抗扰动，让词向量发生微小改变。假设Nezha模型得到的字嵌入表示为X，需要预测的标签label为Y，则对抗扰动策略的具体实现如下：

$$\min_{\theta} E_{(X,Y)\in D}\left[\max_{\Delta X\in\Omega} Loss(X + \Delta X, Y; \theta)\right] \tag{9}$$

对字向量表示X加入对抗扰动ΔX，使得Nezha的损失值增大，但同时受限于约束空间Ω，对抗样本$X + \Delta X$在构建完成之后，输入到Nezha模型通过最小化模型的损失值来更新参数。实验过程中采用快速梯度方法FGM(Fast Gradient Method)计算字嵌入矩阵的梯度ΔW_E，然后再根据得到的梯度对字嵌入矩阵W_E进行对抗扰动。输入序列通过已被对抗扰动的字嵌入矩阵获得新的字向量表示，新的字向量表示用作原字向量表示的对抗样本$X + \Delta X$：

$$\Delta W_E = \epsilon \frac{\nabla_{W_E} Loss(X, Y, \theta)}{||\nabla_{W_E} Loss(X, Y, \theta)||} \tag{10}$$

$$W_E = W_E + \Delta W_E \tag{11}$$

其ϵ是一个超参数，本文实验过程中取值0.5，同时对梯度进行标准化，防止计算出来的梯度过大。

3.4 指数移动平均策略

指数移动平均(Exponential Moving Average, EMA)作为一种深度学习模型常用的调优技巧，可以有效提高模型的性能和鲁棒性。指数移动平均是移动平均的一种，还有简单移动平均(Simple Moving Average, SMA)、权重移动平均(Weight Moving Average, WMA)，主要区别在于平均值的计算方式不一样。指数移动平均是对先前所有数据做加权平均，加权的权重系数呈指数衰减。假设有n组数据$[p_1, p_2, p_3, \ldots, p_n]$，对于简单移动平均来说，计算公式为(12)，对所有样本取平均值。指数移动平均计算公式如(13),其中β表示加权权重值，x_{t-1}是前t-1条的平均值。

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{12}$$

$$x_t = \beta \cdot x_{t-1} + (1-\beta) \cdot p_t \tag{13}$$

训练过程中，神经网络通过对包含标签的样本训练集拟合，神经网络的参数通过最小化训练损失函数来优化。指数移动平均策略在深度学习中上式的p_t相当于在第t次更新得到的所有参数权重，而x_t则是第t次更新的所有参数移动平均数，β表示权重参数。正常神经网络的参数权重相当于一直累积更新整个训练过程的梯度，使用指数移动平均策略的参数权重相当于使用训练过程梯度的加权平均，由于神经网络刚开始训练时不稳定，这时候给予它的加权值较小更为合理，因此在训练神经网络过程中采用指数移动平均策略可以使得模型在测试数据上更健壮，实验过程中衰减率设置为0.999。

4 实验与结果

4.1 数据集描述

本文实验所使用的数据集来自于哈尔滨工业大学组织的CCL2023电信网络诈骗案件分类评测提供的数据集。数据集的案件文本内容是受害人的笔录，是对真实案件的简述，并且此次数据集对案件中原有的一些涉及个人隐私以及敏感信息做了脱敏处理，去除了受害人的姓名、出生日期、地址、社交账号以及银行卡号等信息。此次评测任务通过codalab平台提供技术支持，数据中总共包含12个类别，训练集包含82210条有标签样本供选手自由使用，10276条数据用于线上评测，具体类别及其对应数量如下表1所示。

类别名称	样本数量
刷单返利类	28367
冒充电商物流客服类	11018
虚假网络投资理财类	9469
贷款、代办信用卡类	8883
虚假征信类	6771
虚假购物、服务类	5647
冒充公检法及政府机关类	3651
冒充领导、熟人类	3525
网络游戏产品虚假交易类	1723
网络婚恋、交友类（非虚假网络投资理财类）	1324
冒充军警购物类	874
网黑案件	958
总计数目	82210

Table 1: 标签类别及其数量统计

本次实验所用数据集的长度及其数量分布如下图3所示，由图可见数据集的最长长度能达到1000以上，根据分析数据的最大长度为1865，平均长度为362，1/2的数据长度在309，3/4的数据长度都达到了437。在神经网络的训练过程中，主要是通过矩阵乘法运算，在运算过程中同一个批次的数据如果长度不一样会导致运算出错。因此对输入文本进行词向量化过程中需要保

证输入同一批次文本长度一样，如果对输入文本长度较短，模型可能无法很好理解文本所要表达的意思，导致分类精度较低；而选取的长度过长，会导致短文本填充过多无用字符，导致训练速度比较慢。在实验过程中，我们对选取的文本长度设置为512，并且采用动态填充的方式，将一个批次中的数据长度填充到当前批次中数据的最大长度，当长度过长时就取前512个字符。

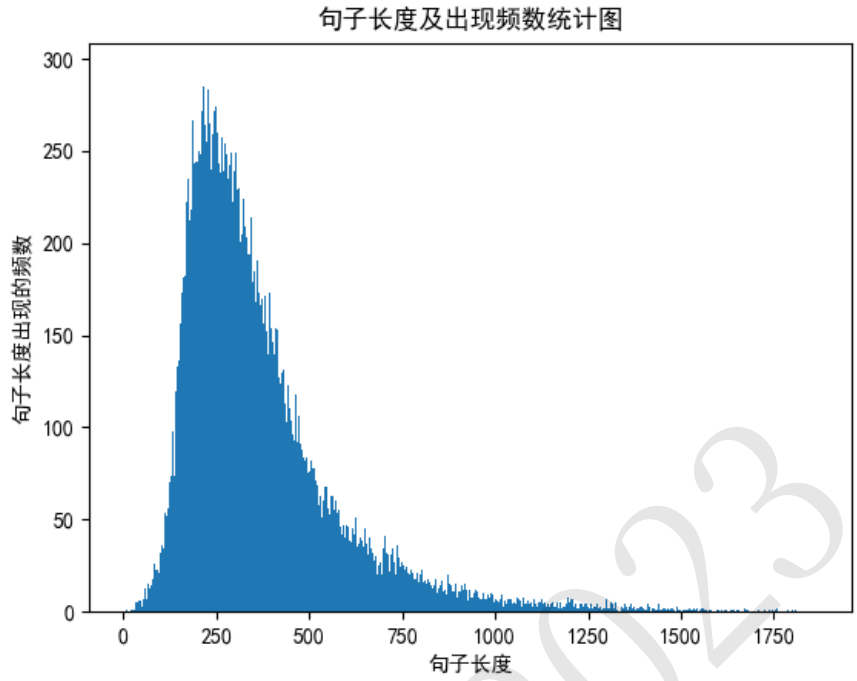

Figure 3: 电信网络诈骗数据集的长度分布图

4.2 评测指标

本文使用的评测指标是macro-f1，这也是作为多类别文本分类任务常见的评价指标。macro-f1同时兼顾了分类模型的精确率和召回率，可以看作是模型精确率和召回率的一种加权平均，最大值是1，最小值是0，值越大表示效果越好。在多分类任务中，评价模型性能有两种f1-score，分别是micro-f1和macro-f1，其中micro-f1计算过程中，每一个样本的权重都相同；macro-f1计算过程将每一类别的权重视为相同，macro-f1计算公式为：

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{16}$$

$$macro - f1 = \frac{1}{n} \sum_{i=1}^{n} f1 - score_i \tag{17}$$

公式中P和R分别表示精确率和召回率，其中公式(17)计算过程中n表示数据的类别总数，本次任务中n=12，下标i表示类别属于第i类的f1值，由上述公式计算每一个类别的f1值，然后再求和取平均得到macro-f1值。

4.3 参数设置

本文使用的预训练模型权重是华为开源的Nezha，输入的句子最大长度设置为512，训练轮数为5，批大小为16，采用差分学习率。在训练过程中也对预训练模型和全连接层参数设置不同学习率，对Nezha模型参数的学习率设置为3e-5，线性层参数学习率则设置较大，为3e-3，同时训练过程中也结合学习率预热，在预热期间，让学习率从0线性增加到优化器中的学习率3e-3，之后让学习率从优化器中的初始学习率线性降低到0，优化器采用AdamW，通过交叉熵损失函数优化更新模型参数。

4.4 实验结果分析

基于此次任务数据集，官方给出了相应的基线分数，分别是基于TextCNN模型和基于Bert微调的分类模型，两者的macro avg f1分别是0.8464和0.8503。本文实验将训练数据划分为10份，其中1份用于训练过程中检验效果，9份用于训练，并且训练每完成一轮对验证集进行验证，同时保存验证集上分数最高的模型权重，便于后续对测试集预测。最开始使用预训练语言模型Nezha实验，在测试集上的分数为0.8530相比于Bert模型高出0.0027，比基于TextCNN的分类模型高出0.0066。Nezha模型是基于Bert在预训练任务和结构上做了相应改进，所以在下游任务上会普遍好于Bert模型，而TextCNN模型仅基于卷积神经网络搭建分类模型，其编码能力不如基于Transformer的Nezha模型，同时也没有预训练学习先验知识，其分类效果较差。使用对抗扰动策略的模型比没有使用该策略在测试集评测时macro-f1分数高0.0056，见表2，因为对抗扰动策略通过对模型的embedding层进行对抗扰动来产生对抗样本，模型在训练时收到对抗样本的攻击可以在一定程度上提高模型的鲁棒性，从而提高模型表现能力的目的。此外在对抗扰动策略基础上加入指数移动平均策略，macro-f1分数提高0.0033，权重滑动平均是提供训练稳定性的有效方法，通过滑动平均可以提高模型的泛化性能。基于上述两个策略模型性能得到一定提升之后，将全部数据用于训练，最后训练完对测试集预测，分数为0.862466，为最终测试集分数，整体实验结果如表2所示。

模型名称	macro-f1
textcnn	0.8464
bert	0.8503
nezha	0.8530
nezha+对抗扰动策略	0.8586
nezha+对抗扰动策略+指数移动平均策略	0.8619
nezha+对抗扰动策略+指数移动平均策略+全量数据	0.8625

Table 2: 实验结果

5 结论

本文阐述预训练语言模型运用在电信网络诈骗案件分类任务中，提出使用预训练语言模型Nezha对网络诈骗案件数据进行文本编码，其中base模型拥有12层编码层，每一层学习到的语义信息不一样，对原有Nezha模型输出的编码向量进行改进，本文结合12层编码层以及embedding层的向量来共同学习输入文本的语义信息。此外，还使用了对抗扰动策略和指数移动平均策略来提高分类模型的性能表现以及泛化性。最后本文使用的算法模型在评价指标macro-f1为0.8625。从本文的实验效果来看，使用的算法模型可以在电信网络诈骗案件分类任务上取得较好的效果。但是实验过程中仍有改进的地方，如Nezha模型的规模大，参数多，因此训练整个算法模型对算力和时间的消耗较多，希望在以后的工作中可以降低模型的复杂度，同时提升评价指标。

参考文献

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems, California, USA*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy*.

郝婷, 王薇. 2023. 融合bert和bilstm的中文短文本分类研究. 软件工程, 26(03):58-62

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jan Lehečka, Jan Švec, Pavel Ircing, and Luboš Šmídl. 2020. Adjusting bert's pooling layer for large-scale multi-label text classification. In *Text, Speech, and Dialogue: 23rd International Conference, Brno, CR*.

Jingpeng Zhao and Yinglong Ma. 2020. Joint embedding of words and category labels for hierarchical multi-label text classification. arXiv preprint arXiv:2004.02555.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. arXiv preprint arXiv:1909.00204.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Volume 1.Vancouver, Canada.

Shuyang Wang, Bin Wu, Bai Wang, and Xuesong Tong. 2019. Complaint classification using hybrid-attention gru neural network. In *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, Macau, China*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI conference on artificial intelligence*, California, USA*.

田园, 马文. *2020.* 基于*attention-bilstm* 的电网设备故障文本分类. 计算机应用, *40(S2):24-29.*

Y. Kim. 2014. Convolutional neural networks for sentence classification. arXiv:1408.5882.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223.

张小为, 邵剑飞. *2021.* 基于改进的*bert-cnn*模型的新闻文本分类研究. 电视技术, *45(07):146-150.*

张云翔, 饶竹一. *2020.* 基于*lstm*神经网络的电网文本分类方法. 现代计算机, *2:8-11.*

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.