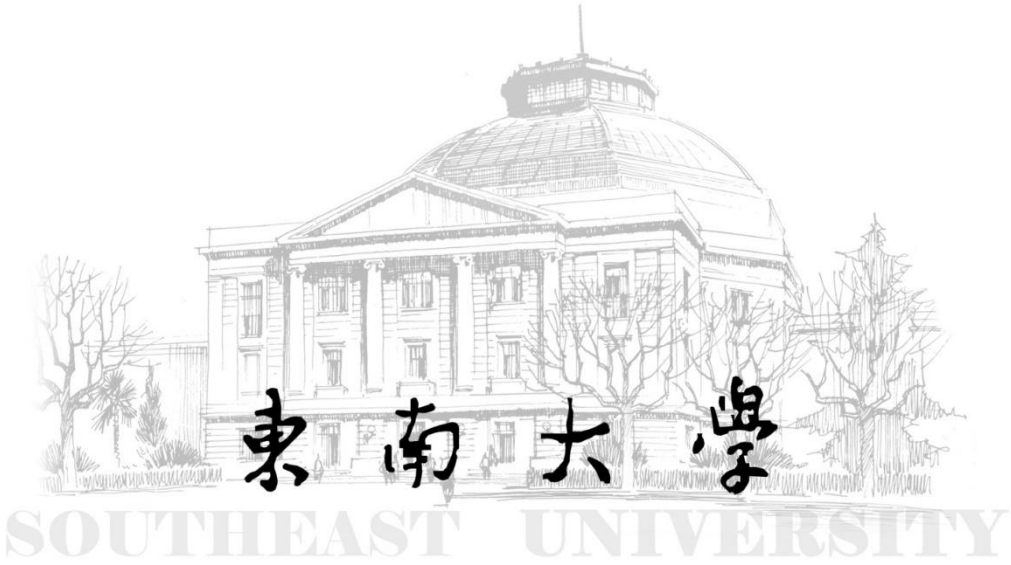

基于 BERT 的
网络恶评分类器的设计与实现

学生姓名 _____学号 _____

教　师 _____

提交日期 _____2022.2.28_____

摘要

互联网已经使其成为了人们生活中不可缺少的一部分，网络社交是现代连接人与人的重要社交方式之一。尽管网络能为人们提供表达自我、获取赞同的渠道，但在海量的交流信息中也暗藏危机：各种恶评在影响着互联网社交用户的使用体验与心理健康。近年来，这个问题引起了学界与工业界的广泛关注。BERT 预训练模型是近年来最强大的自然语言处理模型之一，已有学者证明其在文本分类等自然语言处理任务中具有出色的表现。本技术报告描述了一个基于 BERT 预训练模型的网络恶评分类器的设计与实现过程，包括模型训练与落地，并最终对分类器的分类效果进行了分析与评价。

关键词：文本分类；BERT；网络恶评；自然语言处理

1 背景与意义

1.1 应用背景与意义

随着互联网的飞速发展，网络已经成为了人们日常生活中不可缺少的一部分。社交作为网络的最重要的功能之一，成为了当代人与人之间的重要连接方式，人们通过互联网获取信息、表达自我、发表评论。因此，互联网每天都会产生海量的在线互动信息。尽管网络能为人们提供表达自我、获取赞同的渠道，但在海量的交流信息中也暗藏危机：网络喷子留下的大量恶评正在影响社交媒体用户的使用体验与心理健康。网络恶评来源于人与人之间的在线交互，它不仅仅是一种语言暴力，还是粗鲁的、不尊重的评论，更是人身攻击、网络骚扰和欺凌行为的体现[1]。然而，这种现象在当今的网络社交环境中屡见不鲜。

解决这个问题的关键就是实现网络恶评的精准判断与分类，近年来该问题越来越受到工业界和学术界的重视。2017 年谷歌和 Jigsaw 启动了 Perspective 项目(www.perspectiveapi.com)，该项目基于机器学习模型将评论转化为每种恶评的概率，实现网络恶评的精准实时识别，便于运营者对网站评论进行筛选与净化；同年年底，Jigsaw 在 Kaggle 平台上发布了 Toxic Comment Classification Challenge，旨在集思广益设计高精度的网络恶评分类器。

1.2 技术背景

恶性评论识别与分类本质上是一种文本分类任务，而文本分类是自然语言处理中经典任务之一，它可以被简单定义为：给定文档集合 D 以及一系列文档类别（或标签）的集合 C，定义函数 F 为文档集合 D 中的每份文档分配一个集合 C 中的标签值[2]。

文本分类技术大致可分为文本预处理、文本表示和文本分类模型构建三个阶段[3]。文本预处理是自然语言处理中的首要任务，其核心目标是文本数据去噪，提升数据质量，具体步骤和实际的数据情况、语言以及分类模型的种类密切相关，一般而言包含文本清洗、分词、拼写纠正、剔除停用词等步骤。

文本表示是将文本非结构化数据转化为词嵌入的形式，方便计算机运算与对语义的理解[4]。较为传统的文本表示方式有：考虑词频的共现矩阵法（包括特征降维）、考虑词频与词的稀有度的 TF-IDF 法等，但由于传统方法较为依赖于统计信息，大多忽略了上下文信息以及词语之间的相似性，形成的词嵌入缺乏语义信息，对最终任务的效果较为有限。针对传统方法的局限性，学者们提出了 Word2Vec[5]、GloVe[6]、ELMo[7]、BERT[8]等基于深度学习的预训练词嵌入模型，利用大规模语料库进行训练，这些模型能在不同程度上提取局部的语义信息、语法信息，从单词层级、字符层级或子词层级进行建模。

文本分类模型是在文本表达的基础上选择合适的分类算法实现分类。词袋模型（Bag of words）结合传统机器学习算法是最经典的文本分类算法之一，它假设一篇文档是单词的无序集合，词序信息被忽略。词袋模型通常在学术界与工业界以基线模型的形式存在，比如 Armand Joulin 等人提出了一个高效的文本分类基线模型：该方法用子词层级进行文本表达，使用 n-gram 的方法捕捉局部的词序特征，通过两层前馈神经网络实现分类[9]。目前较为常用的方法是基于卷积神经网络、循环神经网络、自注意力机制的深度学习模型[10]，如 TextCNN[11]，RCNN[12]，BERT[8]等。大量实验可以证明，基于深度学习模型的分类效果整体上优于传统的分类方法[13]。

2 需求目标与总体方案设计

本次技术报告旨在描述实现一个类似于 Perspective 的网络恶评识别分类器的过程，其功能可以描述为：输入一段网络评论（英文），输出该评论具有以下六种性质的可能性：恶意（toxic）、穷凶极恶（severe toxic）、猥琐（obscene）、恐吓（threat）、侮辱（insult）以及种族歧视（identity hate）。

总体方案设计如图 2-1 所示。方案分为“模型训练”与“模型落地”两个部分，其中前者是分类器的核心，后者是在前者的基础上进行包装与落地。

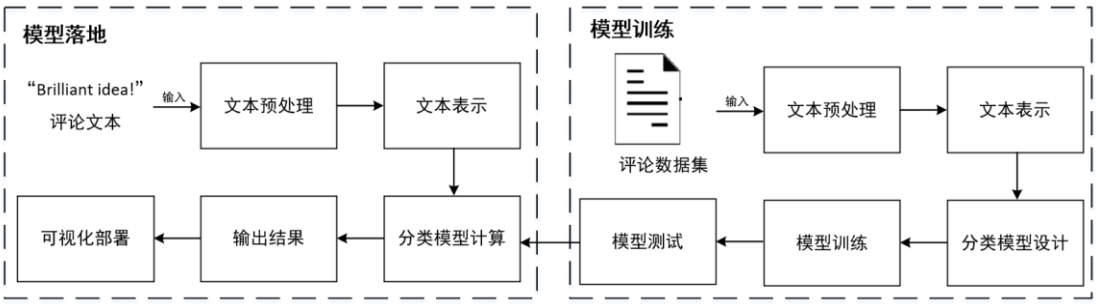

图 2-1 总体方案示意图

在模型训练阶段，基于 Kaggle 的 Toxic Comment Classification Challenge（https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data）提供的维基百科评论数据，经过文本预处理去除与恶意评论分类无关的字符，使用 WordPiece[14]嵌入模型进行文本表示，借助 huggingface 提供的 BERT 预训练模型，采用早停和学习率衰减机制进行模型微调，最终由列平均 AUC 进行模型精度的评价。

在模型落地阶段，基于 Python 的 Gradio 库进行简单可视化部署，使用与训练阶段一致的模型预处理方法、文本表示方法，将外部输入的评论文本在训练好的分类模型中进行运算，输出可视化结果。

3 详细设计

3.1 模型训练阶段

3.1.1 数据情况

模型训练采用维基百科评论数据，以人工分类的结果作为真值。其中包含大量良性或中性评论数据没有恶性标签，以及少量恶评数据被打上恶意（toxic）、穷凶极恶（severe toxic）、猥琐（obscene）、恐吓（threat）、侮辱（insult）以及种族歧视（identity hate）中一种或多种标签。

Kaggle平台提供现成的训练集与测试集，其中训练集包含159571条评论及标签，数据分布如表3-1所示。由于正样本比例较低，存在类别不均衡等问题，仅仅使用准确率（accuracy）作为模型精度评价标准并不合理。而AUC同时考虑了分类器对正例和负例的分类能力，在类别不均衡的情况下，依然能对分类器做出合理评价，因此Toxic Comment Classification Challenge主办方最终要求采用列均AUC值（即每一个标签对应AUC的平均值）作为评价模型的唯一指标。

表3-1 训练集数据分布情况

	数据量	占比
无标签文本	143346	89.83%
toxic	15294	9.58%
severe_toxic	1595	1.00%
obscene	8449	5.29%
threat	478	0.30%
insult	7877	4.94%
identity_hate	1405	0.88%

测试集包含153164条无标签的评论数据。为了防止作弊行为，由Kaggle平台统一对测试集的预测结果进行抽样，计算AUC值作为模型在测试集上的表现。

3.1.2 数据预处理

基于对数据的观察，使用正则表达式分别对文本数据进行如下清洗：去除文本数据中的IP地址、去除对分类没有意义的特殊符号（如·@|+[]等）、去除

URL/html标记、去除时间戳等操作。对文本进行预处理后，使用WordPiece模型对数据进行分词，最终得到的长度分布情况如图3-1与表3-2所示，其中图3-1中的红线是85%分位。可以见得，整体上评论文本长度适中，但存在少数极端值。

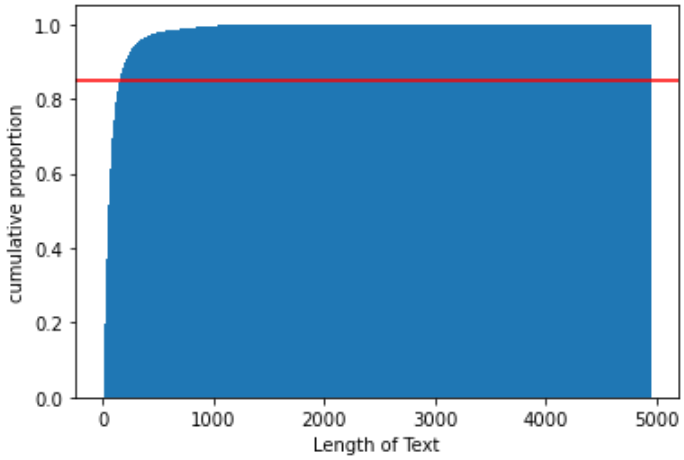

图3-1 长度分布累积比例曲线

表3-2 长度分布情况

统计量	值	统计量	值
均值	95.02	50%分位	51
最小值	2	75%分位	103
最大值	4950	90%分位	206
标准差	149.73	95%分位	310

3.1.3 模型设计

传统的词袋模型与TextCNN在本数据集上已被前人证明效果较为有限[1]。而BERT模型是一种基于Transformer[15]的预训练模型，它的自注意力机制能针对上下文对单词进行编码，因此可以准确地理解在网络评论中同一单词在不同上语境中的含义，有助于解决利用多义词进行侮辱、人身攻击的情况；同时，由于模型引入了[CLS]标记，在自注意力机制的作用下，使其还具备了对整句进行编码的能力，能够精准把握句子的整体含义。一般情况下，仅添加一个输出层，经过微调即可使BERT模型在不同的任务中具有优异的表现[8]。因此分类模型设计将以BERT预训练模型为基础，采用预训练+微调的方式进行分类模型的训练。

分类器所使用的第一个分类模型内核，如图3-2所示，模型架构与Devlin将

BERT用于情感分析所使用的几乎完全相同，其中BERT模型的内部结构可以参阅相关文献[8]，本文不再赘述。该模型直接将BERT模型中对整句话的嵌入，即[CLS]标记对应的嵌入输入到两层前馈神经网络中，并使用了Dropout机制防止过拟合，最终经过Sigmoid函数后可直接输出对应6类恶评标签的概率。

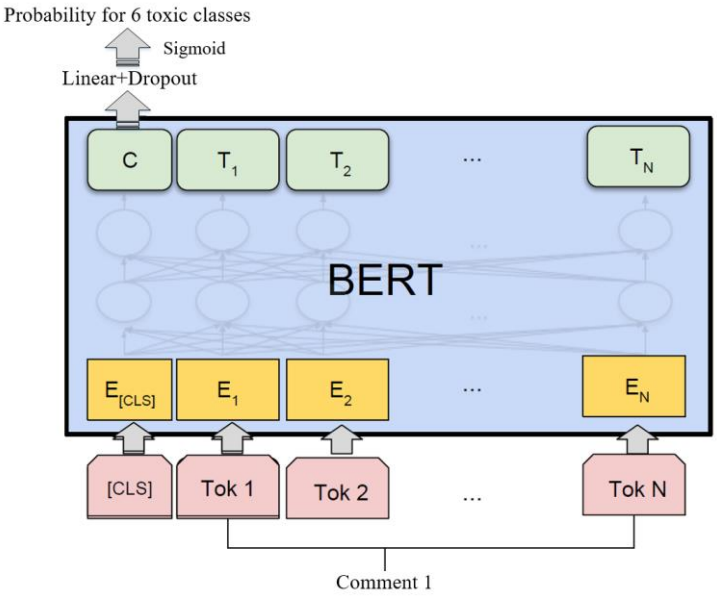

图3-2 微调BERT模型架构

值得注意的是，恶评识别与分类和情感分析之间本质上存在一定的差别：情感分析的目标是把握整句话的情感基调，所以对句子嵌入进行直接分类合乎常理；而网络恶评中某些恶评性质可在短短几个英文单词直接体现，比如“nigga”、“chingchong”等词就可直接体现种族歧视的性质等，因此仅对整句话的嵌入进行分类挖掘可能引入一定的偏差。针对恶评分类的这一特定，采用了BERT-CNN模型作为分类器所使用的第二个内核。该模型在BERT的基础上用TextCNN对每个n-gram进行进一步挖掘，如图3-3所示。该模型将每个词对应的BERT输出直接输入TextCNN网络，分别经过卷积运算、全局最大池化、特征拼接、前馈神经网络运算，将运算的结果与BERT模型输出的句子嵌入经过线性层后的输出相加，最终经过Sigmoid函数得到对应6类恶评标签的概率。

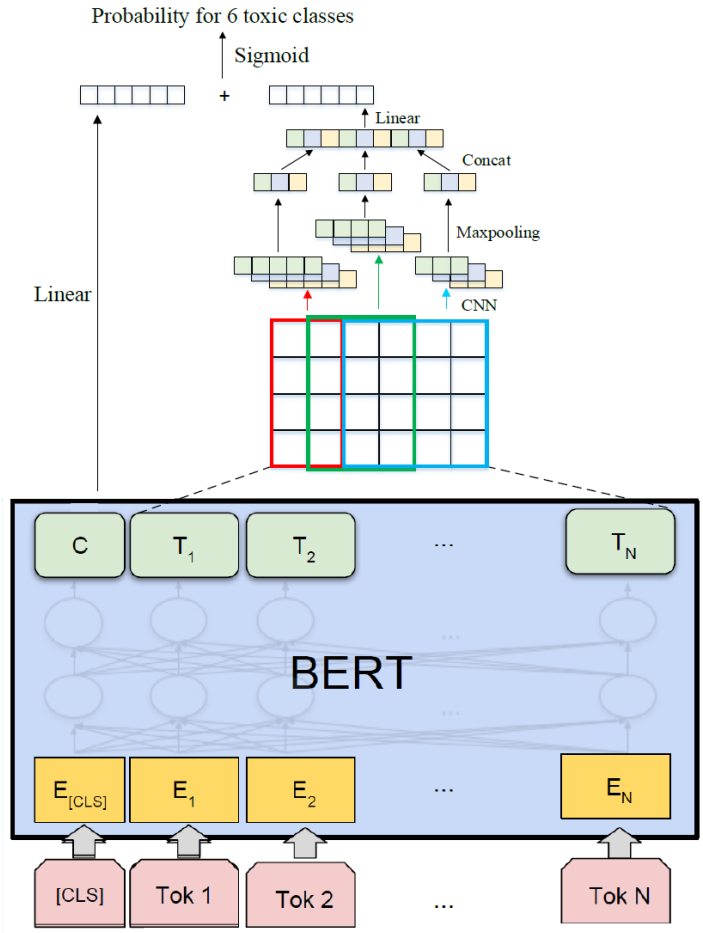

图3-3 BERT-CNN模型架构

3.1.4 训练（微调）方法

模型在Google Colab pro提供的nvidia tesla P100（16G）显卡上进行训练，每批大小为32。由于显存的限制，统一将所有长句仅截取前256个词，结合表3-2中的长度分布情况，这种长度截取方案对绝大部分文本数据没有任何影响。

在训练阶段，先训练数据按照4:1的比例划分为训练集与验证集。由于本次预测任务属于多标签分类任务，因而损失函数选用二分类交叉熵（Binary Cross Entropy）。模型基于反向传播算法，基于AdamW[16]优化器进行参数更新，训练采用了如下的早停策略：模型每训练1200个循环（iteration）进行一次验证，保存在验证集上表现最优的模型、优化器状态以及相应的AUC值。若连续5次验证结果均无法超越之前的最优模型，则舍弃当前优化进度，将模型和优化器重新恢复至之前最优模型的状态，并将学习率折减为原来的1/2。当累计进行了5次学习率衰减时或到达最大训练轮次100轮（epoch）时，模型进行到达早停条

件。

3.1.5 模型测试与分析

模型在Kaggle平台的测试结果如表3-3所示，平台测试抽取了测试集中90%的数据，排名是在所有参赛队伍榜单中的排名情况。

表3-3 模型评价

	AUC	排名
微调BERT模型	0.98452	1539/4540
BERT-CNN模型	0.97563	3191/4540

出乎意料的是，微调BERT模型的表现明显优于BERT-CNN模型。选取了几个典型的评论例句，如表3-4所示，经过初步的分析，可以得出如下结论：

表3-4 BERT与BERT-CNN模型分类对比

	comment		toxic	severe_toxic	obscene	threat	insult	identity_hate
1	Hi! I am back again! Last warning! Stop undoing my edits or die!	真值	1	0	0	1	0	0
		BERT	98.86%	0.55%	2.06%	74.48%	5.18%	0.11%
		BERT-CNN	**16.25%**	0.33%	1.96%	**1.07%**	4.19%	0.16%
2	Stupid peace of shit stop deleting my stuff asshole go die and fall in a hole go to hell!	真值	1	1	1	0	1	0
		BERT	99.98%	39.97%	99.65%	3.21%	98.42%	0.37%
		BERT-CNN	99.98%	21.45%	98.78%	17.50%	97.32%	1.61%
3	And it looks like it was actually you who put on the speedy to have the first version deleted now that I look at it.	真值	0	0	0	0	0	0
		BERT	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
		BERT-CNN	0.54%	0.00%	0.18%	0.00%	0.15%	0.00%
4	Why can't you believe how fat Artie is? Did you see him on his recent appearence on the Tonight Show with Jay Leno? He looks absolutely AWFUL! If I had to put money on it, I'd say that Artie Lange is a can't miss candidate for the 2007 Dead pool! Kindly keep your malicious fingers off of my above comment, . Everytime you remove it, I will repost it!!!	真值	1	0	0	0	0	0
		BERT	90.53%	0.12%	2.51%	0.28%	25.86%	0.52%
		BERT-CNN	**30.95%**	0.11%	1.20%	0.53%	6.68%	0.48%

① BERT-CNN对无恶意的评论（评论3）判断准确，但较BERT模型仍有一定差距，BERT模型能更为肯定地确定句子整体的无恶意。

② BERT-CNN模型对含有敏感词汇的恶评性质分类较为精准，如评论2中，对于“stupid”、“sh*t”、“go to hell”等词模型能够精准将其分类为恶意（toxic）与侮辱（insult），“as**ole”对应猥琐（obscene），即使该句没有威胁之意，但BERT-CNN仍针对于“go die”分类至威胁（threat）标签的可能性较微调BERT大很多。但对于没有用到敏感词的恶评分类效果很差，如评论1和4所示，两句虽没有用到敏感词汇，评论1中“last warning”、“die”以及评论4的字里行间都有带有恶意。

③ BERT-CNN对于恶评的理解似乎只浮于表面，更关注某一个或几个词，而对句意的整体把握不足。

推测原因可能是BERT模型本身网络较深，挖掘到的语义信息较为充足，而TextCNN在最大池化的过程中使其大量信息丢失，因此卷积神经在这个任务中可能并不适合直接与BERT的输出相连接。

3.2 模型落地阶段

可视化部署基于python的Gradio库对算法进行封装与落地，内置与训练阶段一致的模型预处理方法、文本表示方法、两个分类模型内核，使用界面如图3-4所示。在该界面中可直接在“COMMENT TEXT”框内输入评论，在“CLASSIFIER KERNEL”选择合适的分类模型内核，提交后等待结果在“OUTPUT”中以DataFrame的形式呈现。

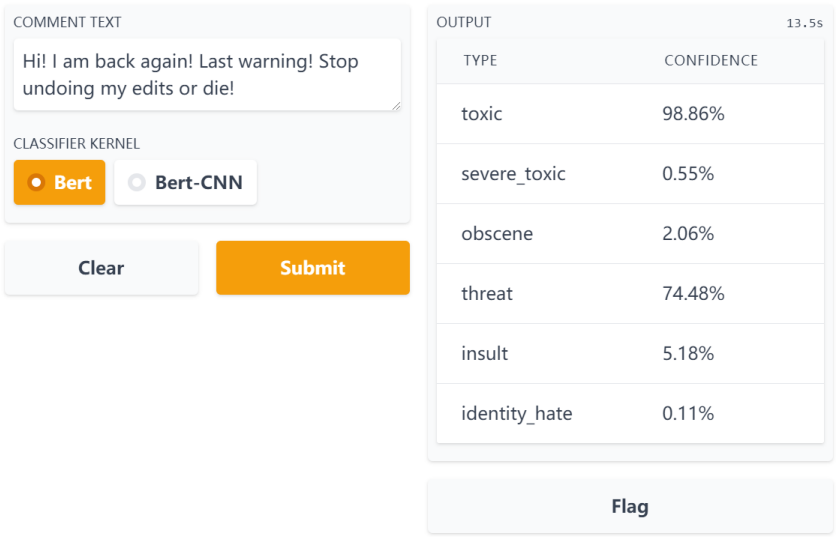

图3-4 使用界面示意

目前的部署尚存在运算速度远慢于不进行封装时的计算速度、输入新的评论进行分类前必须点击“Clear”等问题，推测原因可能和Gradio库内部封装的运行机制有关。

4 总结

本次技术报告从背景与意义、需求目标与总体方案设计、详细设计等三个部分详细描述了一个端到端的基于BERT的网络恶评分类器设计实现的全过程，其中仍有诸多不完善之处，无法直接投入到现实应用中，仍需要后续深入思考与探究。

参考文献

[1] GEORGAKOPOULOS S V, TASOULIS S K, VRAHATIS A G, et al. Convolutional Neural Networks for Toxic Comment Classification; proceedings of the 10th Hellenic Conference on Artificial Intelligence, Patras, Greece, F, 2018 [C]. Association for Computing Machinery.

[2] MURTY M R, MURTHY J, PVGD P R. Text document classification based-on least square support vector machines with singular value decomposition [J]. International Journal of Computer Applications, 2011, 27(7): 21-6.

[3] 蔡梦梦. 基于改进的 DenseNet 短文本分类算法研究 [D]; 南京邮电大学, 2021.

[4] 吴佳君. 面向文本分类任务的深度学习方法研究 [D]; 南京信息工程大学, 2021.

[5] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality; proceedings of the NIPS, F, 2013 [C].

[6] PENNINGTON J, SOCHER R, MANNING C D. Glove: Global vectors for word representation; proceedings of the the 2014 conference on empirical methods in natural language processing (EMNLP), F, 2014 [C].

[7] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations; proceedings of the NAACL, F, 2018 [C].

[8] DEVLIN J, CHANG M-W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:04805, 2018,

[9] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification [J]. arXiv preprint 160701759, 2016,

[10] 何力, 郑灶贤, 项凤涛, et al. 基于深度学习的文本分类技术研究进展 [J]. 计算机工程, 2021,

47(2): 1-11.
[11] KIM Y. Convolutional Neural Networks for Sentence Classification; proceedings of the EMNLP, F, 2014 [C].
[12] LAI S, XU L, LIU K, et al. Recurrent convolutional neural networks for text classification; proceedings of the Twenty-ninth AAAI conference on artificial intelligence, F, 2015 [C].
[13] Y. Z. A Review of Text Classification Based on Deep Learning; proceedings of the the 2020 3rd International Conference on Geoinformatics and Data Analysis, F, 2020 [C].
[14] WU Y, SCHUSTER M, CHEN Z, et al. Google's neural machine translation system: Bridging the gap between human and machine translation [J]. arXiv:160908144 2016,
[15] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need; proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, F, 2017 [C].
[16] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization [J]. arXiv preprint arXiv:171105101, 2017,