doi: 10.3969/j.issn.0490-6756.2020.03.009

基于卷积神经网络和自注意力机制的文本分类模型

汪嘉伟, 杨煦晨, 琚生根, 袁 宵, 谢正文
(四川大学计算机学院, 成都 610065)

摘 要: 单词级别的浅层卷积神经网络(CNN)模型在文本分类任务上取得了良好的表现. 然而,浅层 CNN 模型由于无法捕捉长距离依赖关系,影响了模型在文本分类任务上的效果. 简单地加深模型层数并不能提升模型的效果. 本文提出一种新的单词级别的文本分类模型 Word-CNN-Att,该模型使用 CNN 捕捉局部特征和位置信息,利用自注意力机制捕捉长距离依赖. 在 AGNews、DBPedia、Yelp Review Polarity、Yelp Review Full、Yahoo! Answers 等 5 个公开的数据集上,Word-CNN-Att 比单词级别的浅层 CNN 模型的准确率分别提高了 0.9%、0.2%、0.5%、2.1%、2.0%.

关键词: 文本分类; 卷积神经网络; 自注意力机制; 长距离依赖
中图分类号: TP391 **文献标识码**: A **文章编号**: 0490-6756(2020)03-0469-07

Text classification model based on convolutional neural network and self-attention mechanism

WANG Jia-Wei, YANG Xu-Chen, JU Sheng-Gen, YUAN Xiao, XIE Zheng-Wen
(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: The word-level shallow convolutional neural network (CNN) model has achieved good performance in text classification tasks. However, shallow CNN models can't capture long-range dependencies, which affects the model's performance in text classification tasks, but simply deepening the number of layers of the model does not improve the model's performance. This paper proposes a new word-level text classification model Word-CNN-Att, which uses CNN to capture local features and position information, and captures long-range dependencies with self-attention mechanism. The accuracy of Word-CNN-Att, on the five public datasets of AGNews, DBPedia, Yelp Review Polarity, Yelp Review Full, Yahoo! Answers, is 0.9%, 0.2%, 0.5%, 2.1%, and 2.0% higher than the word-level shallow CNN model respectively.

Keywords: Text classification; Convolutional neural network; Self-attentionmechanism; Long-range dependencies

1 引 言

文本分类为自由文本文档分配预定义的类别,是自然语言处理领域的基础性任务. 文本分类的应用包括情感分析[1]、问题分类[2]、主题分类[3-5]等. 卷积神经网络(Convolutional Neural Network, CNN)[6-8]广泛应用于文本分类. 单词级别的浅层 CNN 模型[6]使用预训练的词向量[9]作为输入,利

收稿日期: 2019-11-04
基金项目: 2018 年四川省新一代人工智能重大专项科技项目(2018GZDZX0039)
作者简介: 汪嘉伟(1993-), 男, 硕士研究生, 研究方向为自然语言处理.
通讯作者: 琚生根. E-mail: jsg@scu.edu.cn

用多种具有不同过滤器的 CNN 抽取文本序列的局部特征,在文本分类任务上取得了良好的表现. 由于模型的深度较浅(只有一层 CNN),单词级别的浅层 CNN 模型无法捕捉长距离依赖[10]. 文献[11]详细研究了 CNN 模型的深度对分类效果的影响,发现对于单词级别的 CNN 模型,加深模型的层数并不能提高模型的准确率,反而导致模型准确率的下降. 为了捕捉长距离依赖,本文引入自注意力机制. 首先,文本序列中的每个单词通过 CNN 得到一个上下文表示,自注意力机制通过计算所有单词的上下文表示两两之间的相似度捕捉长距离依赖;然后,利用最大池化得到文本序列的最终表示;最后,将该表示送入全连接层得到分类结果. 与单词级别的浅层 CNN 模型比较,本文的模型在 AGNews、DBPedia、Yelp Review Polarity、Yelp Review Full、Yahoo! Answers 5 个公开的数据集上准确率得到了一致的提升.

2 相关工作

在文本分类上,有着大量的研究. 传统的方法使用线性模型[4]或支持向量机[12-13]根据手工构造的文本特征对文本进行分类. 这些特征包括词袋特征、n-gram 特征、TF-IDF 特征等.

近年来,随着深度学习的发展,神经网络模型广泛应用于文本分类[1,6-8,14-16]. 文本分类任务的神经网络模型主要分为三大类:基于 RNN 的模型、基于 CNN 的模型和基于注意力机制的模型.

RNN 适用于处理序列输入,因此,许多 RNN 的变种被应用于文本分类. 文献[14]利用 LSTM 建模序列,文献[1]利用 LSTM 和门控 RNN 建模句子间的关系. 文献[15]利用层级 GRU 对文档进行建模,并利用注意力机制捕获文档中重要的单词信息和句子信息. 文献[16]将残差连接[17]引入 RNN,使模型能够处理更长的序列.

CNN 在计算机视觉领域获得了巨大的成功[17-18],文献[19]首次将 CNN 应用于自然语言处理任务. 文献[6]使用预训练的词向量[9]作为输入,利用一层 CNN 捕捉文本序列的局部特征和位置信息. 文献[7]首次探索了字符级别的深层 CNN(6 层)分类模型. 文献[8]构建了一个字符级别的极深的 CNN(29 层)分类模型.

文献[20]首次仅利用注意力机制解决自然语言处理任务,没有使用任何 RNN 和 CNN 结构. 文献[21]将注意力机制应用于文本分类任务,与文献[20]相同,没有使用任何 RNN 和 CNN 结构.

基于 RNN 的分类模型受制于 RNN 的串行结构,无法在序列上并行计算. 字符级别的深层 CNN 模型由于模型深度的急剧增加,导致模型的计算复杂度随之上升,严重影响了模型在实践中的应用. 仅仅基于注意力机制的模型无法捕捉文本序列的局部特征. 单词级别的浅层 CNN 模型无法捕捉长距离依赖. 本文结合 CNN 和自注意力机制,提出一种新的单词级别的文本分类模型 Word-CNN-Att. Word-CNN-Att 使用 CNN 捕捉文档的局部特征,利用自注意力机制捕捉长距离依赖.

3 模 型

模型的整体架构如图 1 所示. 模型由卷积层、自注意力层、池化层和全连接层组成. 卷积层用于捕捉文本序列的局部特征和位置信息,自注意力层用于捕捉长距离依赖,池化层用于获得文本序列的最终表示,全连接层用于最后的分类.

3.1 卷积层

卷积层用于提取输入序列的局部特征和位置信息. $x_i \in R^d$ 是一个 d 维的向量,表示输入序列中的第 i 个单词,一个长度为 n 的序列表示为:

$$x_{1,n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n \tag{1}$$

其中,$\oplus$ 是一个连接操作符;$x_{i,i+j}$ 表示单词 x_i,x_{i+1},…,x_{i+j} 的连接;过滤器 $w \in R^{k \times d}$ 对具有 k 个单词的窗口进行卷积操作,产生新的特征. 例如,特征

$$c_i = f(w \cdot x_{i-(k-1)/2,i+(k-1)/2} + b) \tag{2}$$

其中,$b \in R$ 是偏置;f 是非线性函数 ReLU. 对于超过序列边界的索引,本文采用零填充. 这个过滤器应用到每个可能的窗口,产生一个特征图.

$$\hat{c} = (c_1, c_2, \cdots, c_n) \tag{3}$$

卷积层共有 m 个核宽为 k 的过滤器,对每个过滤器重复上述过程,并将得到的特征图连接起来,得到:

$$Z = (z_1, z_2, \cdots, z_n) \tag{4}$$

$Z \in R^{n \times m}$. 如图 1 所示,本文采用多个核宽分别为 3、4、5 的过滤器.

3.2 自注意力层

自注意力机制的核心在于点乘注意力[20],点乘注意力的计算过程如图 2 所示,定义如下.

$$Attention(Q,K,V) = soft \quad \max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

其中，Q,K,V 分别表示“查询”、“键”和“值”；d_k 是缩放因子，表示 K 的维度. 对于较大的 d_k 值，点乘的积过大，从而将 softmax 函数推向具有极小梯度的区域[20]. 为了抵消这种影响，利用 $\frac{1}{\sqrt{d_k}}$ 缩放点积[20].

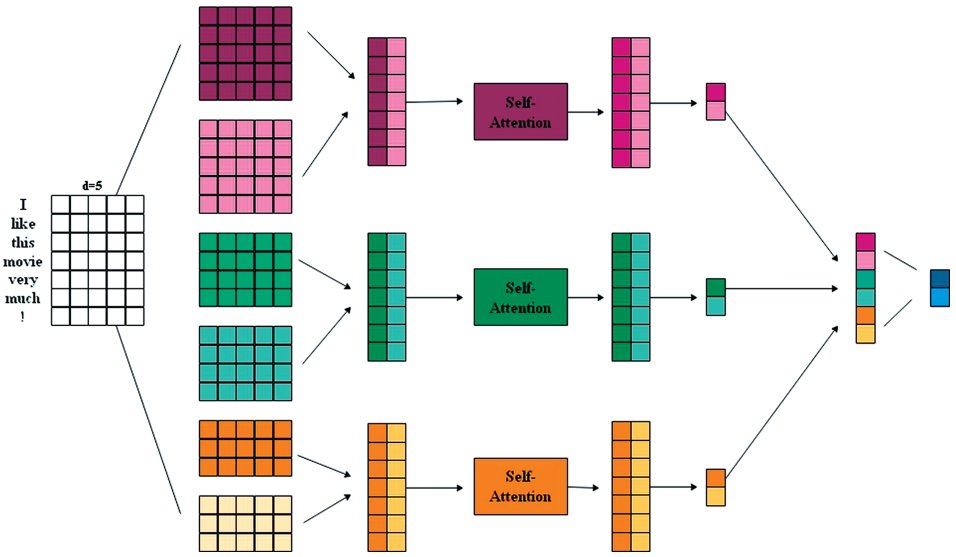

图 1　模型架构. 使用 3 种不同的过滤器，分别具有核宽：3，4，5，每种过滤器有两个

Fig. 1　Architecture of Model. 3 convolutional layers with respective kernel window sizes 3，4，5 are used，and each of which has 2 filters

令 $Z=(z_1,z_2,\cdots,z_n)$ 为卷积层的输出，即自注意力层的输入，$z_i\in R^m$. 在自注意力机制中，Q，K，V 都是同一向量的线性变换. 因此，本文定义自注意力如下.

$$Self-Att(Z)=Attention(ZW_Q,ZW_K,ZW_V)=soft\ \max\left(\frac{ZW_QW_K^TZ^T}{\sqrt{d_k}}\right)ZW_V \tag{6}$$

其中，$W_Q,W_K,W_V\in R^{m\times m}$，$W_Q,W_K,W_V$ 都是模型的参数，在模型训练中学习得到.

自注意力机制通过计算整个序列中所有令牌两两之间的相似度捕捉任意距离的长距离依赖[20]. 与 RNN 不同，由于 RNN 下一时刻的输入依赖于上一时刻的隐藏层状态，所以捕捉距离为 n 的长距离依赖，RNN 的时间复杂度为 $O(n)$. 而自注意力机制可以并行计算任意两两令牌之间的相似度，捕捉距离为 n 的长距离依赖的时间复杂度为 $O(1)$. 因此，自注意力机制有着非常好的并行性.

自注意力层的整体结构如图 3 所示，与文献[18]相同，本文引入残差连接[17]和层归一化[22]. 因此，自注意力层的输出为

$$SelfAtt-Out=layernorm(Self-Att(Z)+Z) \tag{7}$$

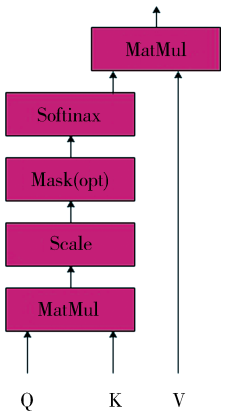

图 2　点乘注意力

Fig. 2　Dot-product attention

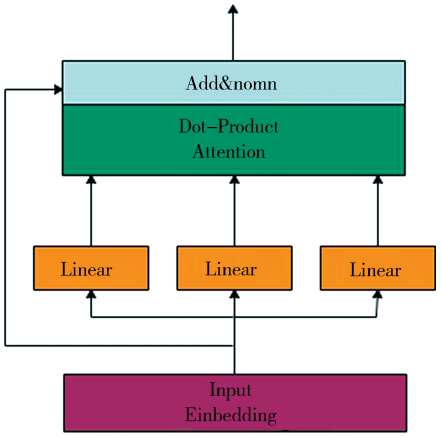

图 3　自注意力层结构

Fig. 3　Architecture of self-attention layer

3.3 池化层与全连接层

对于自注意力层的输出,应用最大池化,每个特征图得到一个最大值,将所有特征图的最大值连接起来,得到输入序列的最终表示.

$$g=(e_1,e_2,\cdots,e_m) \tag{8}$$

最后,一个线性层将 g 映射成文本类别数目的维度.

$$y=W_y g+b_y \tag{9}$$

4 实 验

4.1 任务和数据集

本文实验采用 5 个大规模的文本分类数据集,这些数据集在文献[7]中提出,包括 4 种分类任务:新闻分类、本体分类、情感分析和主题分类. 数据集的具体情况如表 1 所示. 表 1 中,"# Train"代表训练集的样例数目;"# Test"代表测试集的样例数目;"# Classes"代表数据集的种类个数;"# Average Length"代表数据集中样例的平均单词数目.

表 1 数据集的分布

Tab. 1 Statics of datasets

Dataset	# Train/k	# Test/k	# Classes	# AverageLength	Classification Task
AGNews	120	7.6	4	45	English news categorization
DBPedia	560	70	14	55	Ontology classification
Yelp Review Polarity	560	38	2	153	Sentiment analysis
Yelp Review Full	650	50	5	155	Sentiment analysis
Yahoo! Answers	1400	60	10	112	Topic classification

4.2 实验细节

本文使用 NLTK 对语料进行分词,仅使用在训练集中至少出现 3 次的单词构建词表. 词表中未出现的单词使用一个特殊的令牌 UNK 代替.

本文使用斯坦福大学公开发行的 Glove 300 维词向量[9]作为预训练的词向量. 对于未出现在预训练的词向量中的单词,本文使用从均匀分布(−0.1, 0.1)中采样的 300 维向量作为其词向量.

本文使用初始学习率为 0.001 的 Adam 优化算法[23]. batch size 设为 64. 对于每个数据集,实验使用训练集的 10%作为验证集. 本文使用核宽为 3、4、5 的过滤器各 100 个. 在模型的输入层和线性层使用 dropout[24],dropout 的丢弃率为 0.5.

4.3 实验结果与分析

本文使用准确率作为评价指标,准确率越大模型效果越好. 实验结果如表 2 所示.

表 2 模型准确率

Tab. 2 Accuracy of model

Model	AGNews/%	DBPedia/%	Yelp Review Polarity/%	Yelp Review Full/%	Yahoo! Answers/%
bag of words[7]	88.8	96.6	92.2	58.0	68.9
ngrams[7]	92.0	98.6	95.6	56.3	68.5
ngrams TFIDF[7]	92.4	98.7	95.4	54.8	68.5
fastText[25]	92.5	98.6	95.7	63.9	72.3
char-CNN[7]	87.2	98.3	94.7	62.0	71.2
char-CRNN[26]	91.4	98.6	94.5	61.8	71.7
char-VDCNN[8]	91.3	98.7	95.7	64.7	73.4
Discriminative LSTM[14]	92.1	98.7	92.6	59.6	73.7
Self-Attention[21]	92.6	98.7	95.2	64.0	74.1
word-CNN	92.8	98.7	95.6	62.9	72.1
word-CNN-Att	93.7	98.9	96.1	65.0	74.1

表 2 的第 2 行至第 4 行展示了传统的方法的准确率，bag of words[7]模型基于训练集中频率最高的 50 000 个单词构建，ngrams[7]模型基于训练集中频率最高的 500 000 个 n-grams 构建，ngrams TFIDF[7]模型与 ngrams[7]模型相同，但使用 TFIDF 作为特征. 从表 2 可知，传统的方法在 AGNews、DBPedia、Yelp Review Polarity3 个相对小的数据集上表现较好，在 Yelp Review Full、Yahoo! Answers 两个相对大的数据集上表现较差. FastText[25]为文本分类模型提供了一个有力的基线.

char-CNN[7]、char-CRNN[26]、char-VDCNN[8]都是字符级别的 CNN 模型，将字符作为基本输入单位. char-CNN 使用了一个深层的 CNN(6 层). 与 char-CNN 相比，本文的模型 Word-CNN-Att 在 AGNews、DBPedia、Yelp Review Polarity、Yelp Review Full、Yahoo! Answers 5 个数据集准确率分别提高了 6.5%、0.6%、1.4%、3.0%、2.9%. char-CRNN 模型利用 CNN 和 RNN 联合学习文本特征，与 char-CRNN 模型相比，Word-CNN-Att 在 5 个数据集准确率分别提高了 2.3%、0.3%、1.6%、3.2%、2.4%. char-VDCNN 构建了一个极深的 CNN(29 层)，与 char-VDCNN 相比，Word-CNN-Att 在 5 个数据集准确率分别提高了 2.4%、0.2%、0.4%、0.3%、0.7%. 可以看到，尽管 char-VDCNN 远比 Word-CNN-Att 深，Word-CNN-Att 在各个数据集上的准确率仍然均超过了 char-VDCNN 模型. 由上述分析可知，字符级别的模型，无论是纯粹的 CNN 模型或结合 CNN 和 RNN 的模型，尽管模型远比 Word-CNN-Att 深，但表现均不如 Word-CNN-Att. Word-CNN-Att 是单词级别的模型，可以有效地利用单词的语义信息，而字符级别的模型无法利用单词的语义信息.

Discriminative LSTM[14]是一个单词级别的模型，利用 LSTM 作为特征提取器，将输入序列中所有单词的隐藏层状态之和作为文本序列的最终表示. 与 Discriminative LSTM 相比，Word-CNN-Att 在 5 个数据集准确率分别提高了 1.6%、0.2%、3.5%、5.4%、0.4%. 与 LSTM 相比，CNN 能够有效地捕捉局部特征，但无法捕捉长距离依赖，而 Word-CNN-Att 利用自注意力机制捕捉长距离依赖.

Self-Attention[21]模型完全基于自注意力机制，没有使用任何 RNN 和 CNN 结构，利用专门的位置向量编码位置信息. 与 Self-Attention 模型相比，Word-CNN-Att 在 5 个数据集准确率分别提高了 1.1%、0.2%、0.9%、1.0%、0.0%. Word-CNN-Att 不仅使用自注意力机制捕捉长距离依赖，并且使用 CNN 学习文本序列的局部特征，而纯粹的自注意力机制无法学习局部特征；与专门的位置向量相比，CNN 能够更有效地学习位置信息.

如表 2 所示，与单词级别的一层 CNN(Word-CNN)模型相比，Word-CNN-Att 在 5 个数据集准确率分别提高了 0.9%、0.2%、0.5%、2.1%、2.0%. 实验结果表明，自注意力机制有效地捕捉了长距离依赖，弥补了 CNN 无法捕捉长距离依赖的不足，提升了模型处理长文本分类任务的能力. 表 3 展示了一个 word-CNN-Att 分类正确，而 word-CNN 分类错误的样例. word-CNN 或许抽取了一些关键的局部信息，比如：great、lovely，从而将该样例错误分类为 positive. 而 word-CNN-Att 模型可以捕捉长距离依赖，因此可以捕捉到 However 之后的信息，所以将该样例正确分类为 negative.

表 3　样例分析

Tab. 3　Examples of analysis

example	word-CNN	word-CNN-Att	label
The location of Redfin Blues is great! It overlooks the river, and makes for a lovely place to hang out and have a few drinks... However, the food and service is sub par. Each time that I have been here, whether it be an extremely bustling Saturday evening, or a slow weekday afternoon, the service has been poor. The food is average, and is way overpriced. There are many other great places in Pittsburgh to go if you want a yummy fish sandwich. Redfin Blues has the potential to be such a great restaurant! But with the poor service and average food, it has a lot to yearn for.	positive	negative	negative

5　结　论

本文提出了一种结合 CNN 和自注意力机制的文本分类模型 Word-CNN-Att. 该模型利用 CNN 提取文本局部特征和位置信息，利用自注意力机制捕捉长距离依赖. 在 5 个大型公开文本分类

数据集上的实验结果表明,Word-CNN-Att 提升了单词级别的浅层 CNN 模型的效果. 在未来的研究中,计划引入外部知识来进一步增强模型的文本分类能力.

参考文献:

[1] Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification [C]//Proceedings of the 2015 conference on empirical methods in natural language processing. Lisbon, Portugal: Association for Computational Linguistics, 2015.

[2] Zhang D, Lee W S. Question classification using support vector machines [C]//Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. New York, USA: ACM, 2003.

[3] 陈波. 基于循环结构的卷积神经网络文本分类方法[J]. 重庆邮电大学学报:自然科学版, 2018, 30: 705.

[4] Wang S, Manning C D. Baselines and bigrams: Simple, good sentiment and topic classification [C]//Proceedings of the 50th annual meeting of the association for computational linguistics. Jeju Island, Korea: Association for Computational Linguistics, 2012.

[5] 高云龙, 左万利, 王英, 等. 基于集成神经网络的短文本分类模型[J]. 吉林大学学报:理学版, 2018, 56: 933.

[6] Kim Y. Convolutional neural networks for sentence classification [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics, 2014.

[7] Zhang X, Zhao J, Lecun Y. Character-level convolutional networks for text classification [C]//Proceedings of the 2015 Advances in Neural Information Processing Systems. Montreal, Canada: Curran Associates, Inc., 2015.

[8] Conneau A, Schwenk H, Barrault L, *et al*. Very deep convolutional networks for text classification [C]//European Chapter of the Association for Computational Linguistics EACL'17. Valencia, Spain: Association for Computational Linguistics, 2017.

[9] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics, 2014.

[10] Strubell E, Verga P, Belanger D, *et al*. Fast and accurate entity recognition with iterated dilated convolutions [C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017.

[11] Le H T, Cerisara C, Denis A. Do convolutional networks need to be deep for text classification? [C]//Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. NewOrleans, Louisiana, USA: Association for the Advancement of Artificial Intelligence, 2018.

[12] Tong S, Koller D. Support vector machine active learning with applications to text classification [J]. J Mach Learn Res, 2001, 2: 45.

[13] Joachims T. Text categorization with support vector machines: Learning with many relevant features [C]//Proceedings of the 1998 European conference on machine learning. Berlin, German: Springer, 1998.

[14] Yogatama D, Dyer C, Ling W, *et al*. Generative and discriminative text classification with recurrent neural networks [C]//Thirty-fourth International Conference on Machine Learning (ICML 2017). Sydney: International Machine Learning Society, 2017.

[15] Yang Z, Yang D, Dyer C, *et al*. Hierarchical attention networks for document classification [C]//Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. San Diego, California, USA: Association for Computational Linguistics, 2016.

[16] Wang Y, Tian F. Recurrent residual learning for sequence classification [C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas, USA: Association for Computational Linguistics, 2016.

[17] He K, Zhang X, Ren S, *et al*. Deep residual learning for image recognition [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, Nevada, USA: IEEE, 2016: 770.

[18] 凌语, 孙自强. 基于卷积神经网络的乳腺病理图像识别算法[J]. 江苏大学学报:自然科学版, 2019, 40: 573.

[19] Collobert R, Weston J. A unified architecture for

natural language processing: Deep neural networks with multitask learning [C]//Proceedings of the 25th International Conference on Machine learning. New York, USA: ACM, 2008.

[20] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need [C]//Proceedings of 2017 Advances in neural information processing systems. Long Beach, California, USA: Curran Associates, Inc., 2017.

[21] Letarte G, Paradis F, Giguère P, *et al*. Importance of self-attention for sentiment analysis [C]//Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Brussels, Belgium: [s. n.], 2018.

[22] Ba J L, Kiros J R, Hinton G E. Layer normalization [J]. Stat, 2016, 1050: 21.

[23] Kingma D P, Ba J. Adam: a method for stochastic optimization [C]// 3rd International Conference on Learning Representations. San Diego, CA, USA: arXiv Org, 2015.

[24] Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: a simple way to prevent neural networks from overfitting [J]. J Mach Learn Res, 2014, 15: 1929.

[25] Joulin A, Grave E, Bojanowski P, *et al*. Bag of tricks for efficient text classification [C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain: Association for Computational Linguistics, 2017.

[26] Xiao Y, Cho K. Efficient character-level document classification by combining convolution and recurrent layers [J]. arXiv, 2016, 1602: 367.

引用本文格式：

中　文：汪嘉伟，杨煦晨，琚生根，等. 基于卷积神经网络和自注意力机制的文本分类模型[J]. 四川大学学报：自然科学版，2020，57：469.

英　文：Wang J W, Yang X C, Ju S G, *et al*. Text classification model based on convolutional neural network and self-attention mechanism [J]. J Sichuan Univ: Nat Sci Ed, 2020, 57: 469.