基于注意力机制和 CNN 的多标签文本分类模型

杨春霞[1,2] 吴佳君[1,2] 瞿 涛[1] 姚思诚[1]

[1](南京信息工程大学 江苏 南京 210044)

[2](江苏省大数据分析技术重点实验室 江苏 南京 210044)

摘 要 针对目前多标签文本分类模型存在无法充分提取文本语义与标签的相互关系,提出一种基于注意力机制和卷积神经网络(CNN)的多标签文本分类模型。通过多头注意力机制和 CNN 对文本进行建模表示,充分挖掘文本全局和局部的语义特征;结合标签与文本信息进行交互注意力计算,捕捉结合文本内容后标签间的相互关系;使用一种自适应融合策略进一步提取两者语义信息。实验结果表明,该模型相比于其他主流模型能有效提升多标签文本分类效果。

关键词 多标签文本分类 注意力机制 卷积神经网络 文本表示

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2024.03.024

MULTI-LABEL TEXT CLASSIFICATION MODEL BASED ON ATTENTION MECHANISM AND CNN

Yang Chunxia[1,2] Wu Jiajun[1,2] Qu Tao[1] Yao Sicheng[1]

[1](*Nanjing University of Information Science & Technology*, *Nanjing 210044*, *Jiangsu*, *China*)

[2](*Jiangsu Key Laboratory of Big Data Analysis Technology*, *Nanjing 210044*, *Jiangsu*, *China*)

Abstract To address the problem of being unable to fully extract the relationship between text semantics and label in current multi-label text classification, a multi-label text classification model based on attention mechanism and convolutional neural network is proposed. The multi attention mechanism and CNN were used to represent the text, and the global and local semantic features of the text were fully mined. It combined tags and text information to calculate the interactive attention, and captured the relationship between tags after combining the text content. It used an adaptive fusion strategy to further extract the semantic information of the two. Experimental results show that this model can effectively improve the effect of multi label text classification compared with other mainstream models.

Keywords Multi-label text classification Attention mechanism Convolutional neural network Text representation

0 引 言

在当今信息爆炸的时代背景下,内容丰富的海量文本信息充斥着人们的生活,为了有效地进行文本信息管理,便于人们能高效获取自身所需信息,对文本分类技术的要求已经越来越高,其中多标签的文本分类技术显然已成为研究的重中之重。其在信息检索[1]、情感分析[2]、问答任务[3]和主题识别[4]等应用场景中都具有重要作用。多标签文本分类(Multi-Label Text Classification,MLTC)作为自然语言处理(NLP)文本分类的一项重要且具有挑战性的任务,已经得到了广泛的应用研究。其旨在给文本分配与之相关的多个标签,相比于单标签的文本分类来说,多标签文本分类更为复杂。

MLTC 按照文本自身所含内容做出分类,例如,在文献标记任务中,关于 NLP 领域的论文就可能会有“人工智能”和“计算机科学”等标签。目前常见的模型都是基于深度学习的分类方法,相较于传统的机器学习方法需要人工提取特征,深度学习依靠计算机自

收稿日期:2021 - 01 - 14。国家自然科学基金项目(61273229);江苏省青蓝工程项目。**杨春霞**,教授,主研领域:金融大数据分析,自然语言处理。**吴佳君**,硕士生。**瞿涛**,硕士生。**姚思诚**,硕士生。

主学习、提取特征,节省了大量资源,效果也更为优异,展现出了不可比拟的优越性,所以在 MLTC 任务中,更多的研究者都会使用深度学习的方法来处理该任务。现有的主流模型虽有效解决了特征提取、语义关系不明的缺点,但是对如何深层挖掘文本的全局和局部语义关系,以及捕捉标签与标签之间、标签与文本语义的相互关系问题还未得到有效解决。因此,改善模型在 MLTC 任务中的不足,提升分类精度,对当今时代文本信息等分类场景的实际应用已经刻不容缓。

1　相关工作

到目前为止,研究者在研究 MLTC 方法的各个方面都作出了巨大贡献,尤其是基于深度学习的分类方法更是为人所热捧。Kim[5] 第一次尝试将 CNN 用于文本分类中,利用预训练的词向量(如 Glove[6])构造文本表示,然后使用卷积层进行特征映射,接着使用最大池化层提取重要特征,最后接全连接层输送到分类器分类,取得良好效果。Joulin 等[7] 提出的 FastText 模型,通过词之间的 n-gram 进行向量表示,用文本中词的均值向量来构造文档表示,然后使用 Softmax 层进行分类,在性能和精度上都取得不错效果。Liu 等[8] 将深度学习方法应用于 MLTC 任务中,提出了 XML-CNN 模型,该模型结合 CNN 在文本分类中的优势,设计了一个动态池化层来适应多标签的文本分类,并在池化层和输出层之间附加一个隐藏瓶颈层来减少参数、提高学习能力,最终在多个数据集上都获得了很好的效果。Chen 等[9] 提出 CNN-RNN 的分类模型,其中 CNN 用于捕捉文本特征,RNN 用于多标签预测,实验结果表明它可以有效提高模型性能。但上述方法还存在着对全局语义捕捉不全、未考虑标签相关性的缺点。Kurata 等[10] 基于 CNN,同时还将标签之间的共现关系考虑进来,用于初始化最后输出层参数,从而获得标签之间的关系,该方法在没有额外计算开销的情况下获得了很好的分类性能。Zhang 等[11] 则提出一种实用的深度嵌入方法,该方法建立标签图结构挖掘标签的相关性,也获得了较好的结果。Du 等[12] 提出了 EXAM 模型,该模型利用交互机制来计算词与标签的匹配得分,然后将这些分数聚合成每个类的预测,大量实验表明了该模型的有效性。

此外,注意力(Attention)机制的兴起也使之被广泛应用于文本分类任务中。研究者希望模型能够给文本中更重要的单词或句子分配更高的权重,从而提高模型对重要词句的关注度,达到提高学习效率的目的。Attention 机制的引入也让研究者在 MLTC 任务中更好地考虑文本语义和标签之间的相关性。Yang 等[13] 提出了一个新的 Seq2Seq 模型,该模型将 MLTC 任务视为一个序列生成问题,将标签相关性考虑在内,在编码器阶段使用双向 LSTM 计算每个单词的隐层表示,然后使用 Attention 机制让模型关注于对标签贡献更大的词,最后使用 LSTM 进行解码,实验表明了该方法的有效性。You 等[14] 提出一种基于标签树的 AttentionXML 模型,该模型使用双向 LSTM 来捕获单词的长距离依赖关系,使用多标签 Attention 机制捕获文本中与每个标签最相关的部分,构建具有标签信息的文本表示,在实验中取得了不错效果。肖琳等[15] 提出了一种与 AttentionXML 类似的 LASA 模型,其使用双向 LSTM 获得单词的隐层表示,再利用 Attention 机制获得每个标签和文档中单词的匹配得分,为每个标签学习一个文档表示,不仅考虑了标签的相关性,也缓解了 MLTC 中尾标签问题,获得了更好的效果。王浩瀚等[16] 以 Seq2Seq 模型为基础,提出基于多级特征和混合 Attention 机制的分类算法,该算法在编码时使用两层双向 LSTM 捕捉语义信息,使用空洞卷积捕捉词级信息,解码时仍使用 LSTM 获取标签间的相关性,并引入 Attention 机制使得解码时更关注有用信息。但此类模型都受到循环网络的梯度爆炸、梯度消失、计算复杂度高等问题的限制,并且在提取文本上下文语义、标签之间和与文本间的相互关系还有待提升。

基于上述工作,为了进一步提高 MLTC 的性能,本文提出一种基于注意力机制和 CNN 多标签分类模型。首先利用引入位置编码后的文本表示分别在多头自注意力机制和 CNN 上充分提取全局和局部文本语义信息,然后结合标签表示与经过 CNN 得到的文本表示使用交互注意力机制获得和文本内容后的标签之间的相互关系,最后将两者的输出通过一种自适应融合策略进一步提取语义信息,从而实现在 MLTC 任务上分类精度的提升。

2　模型设计

给定训练集$\{(D_1,y_1),(D_2,y_2),\cdots,(D_n,y_n)\}$,其中 D_m代表训练集中第 m 个文本,包含了一系列词$\{w_1,w_2,\cdots,w_q\}$,y_m代表 D_m的类别标签,其中 $y_m\in\{0,1\}^l$,l 为标签总数,对于 MLTC 任务来说,每个文本 D_m都含有两个及以上的标签,将所有标签表示为 $\boldsymbol{C}=\{c_1,c_2,\cdots,c_l\}$的矩阵。本文的模型框架如图 1 所示,其主要包含这几个部分:(1) 引入位置编码,强调文本中单词的位置信息,为后续没有考虑词序信息的多头

注意力机制(Multi-Head Attention)做准备;(2) 分别使用 CNN 和 Multi-Head Attention 对文本进行建模表示,充分提取文本的局部和全局语义信息;(3) 将标签向量矩阵与经 CNN 输出的文本表示相结合做交互注意力计算,深度捕捉标签间以及与文本间的相互关系;(4) 将(2)和(3)的输出使用自适应融合策略进一步提取语义信息,最后进行分类。相比其他基于深度学习的模型,本文模型能够对文本局部以及全局的语义信息做更为全面的提取并充分学习标签间的相互关系,这对于多标签分类来说都是非常有必要的,因为文本其中一个标签通常只注重文本中的局部部分,而全局语义更有助于模型理解文本,而文本本身的内容也决定了其多个标签之间也具有一定的相关性。

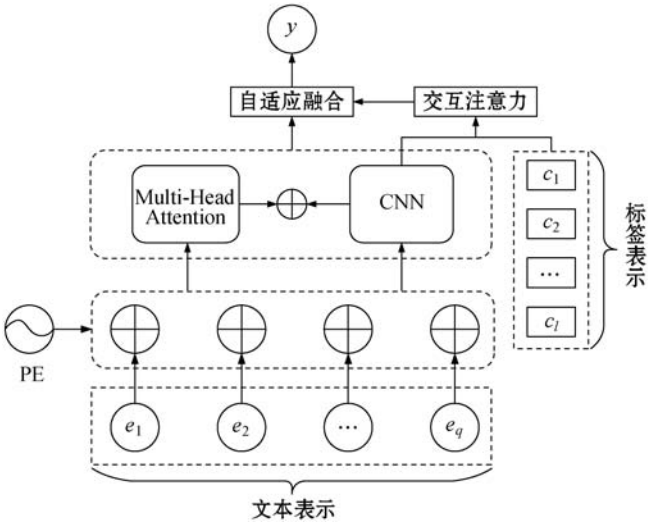

图 1 模型整体框架

位置编码(Positional Embedding)能够为文本中的每个词标记词序信息,由于文本词句的构成都是有顺序的,不同顺序的词构成的语义可能会完全不同,因此强调词语的位置信息是极为必要的,尤其在没有重视词序信息的模型中能够提高学习性能。

2.1 文本与标签的词向量表示

文本词向量与标签词向量均使用 Glove 模型获取词向量表示。Glove 模型基于 Word2Vec[17] 的方法,在语料库上构建了一个词共现矩阵,据此进行词向量的训练,这些词向量能够捕捉到单词间的语义特性,最终能够得到 d 维的词向量表示。在本文模型中,每个词的词向量维度 $\boldsymbol{e}_l \in \mathbf{R}^d$,每个标签的词向量通过标签词的求和平均获得,维度为 $\boldsymbol{c}_l \in \mathbf{R}^d$,由所有标签组成的标签向量矩阵则为 $\boldsymbol{C} \in \mathbf{R}^{l \times d}$。

2.2 位置编码

位置编码通过构建一个与词向量相同维度的矩阵来显示词序信息,其计算公式如下:

$$P_{\mathrm{E}_{(\mathrm{pos},2\mathrm{i})}} = \sin\left(\frac{pos}{10\ 000^{2i/d}}\right) \tag{1}$$

$$P_{\mathrm{E}_{(\mathrm{pos},2\mathrm{i}+1)}} = \cos\left(\frac{pos}{10\ 000^{2i/d}}\right) \tag{2}$$

式中:pos 表示单词的位置索引,i 表示词向量上的维度索引,d 为词向量维度大小,可以看到位置编码在词向量的偶数维度上使用正弦函数,而在奇数维度上使用余弦函数,进而能够获得整个文本词的位置编码矩阵 $\boldsymbol{P} = \{p_1, p_2, \cdots, p_q\}$。最终,引入位置编码后的文本词向量就表示为:

$$\tilde{\boldsymbol{e}} = \{e_1 + p_1, e_2 + p_2, \cdots, e_q + p_q\} \tag{3}$$

2.3 CNN

CNN 凭借其结构简单、计算复杂度低、训练高效等优势广泛运用在文本分类任务中,本文也将使用 CNN 来挖掘文本语义的局部信息。其模型结构如图 2 所示。

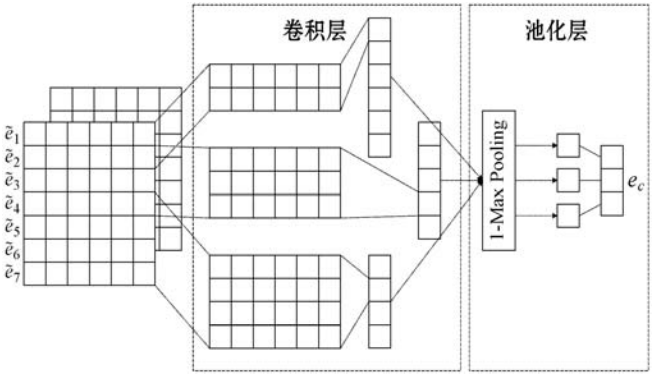

图 2 CNN 模型

首先将引入位置编码后的词向量矩阵 $\tilde{\boldsymbol{e}}$ 输送到卷积层中做特征提取,然后经过最大池化层提取最终的特征并进行拼接,得到整个文本的最终输出 $\boldsymbol{v}_e$,其公式为:

$$v_i = f(\boldsymbol{w} \cdot \tilde{\boldsymbol{e}}_{i:i+m-1} + b) \tag{4}$$

$$v = [v_1, v_2, \cdots, v_{q-m+1}] \tag{5}$$

$$\hat{v} = \max\{\boldsymbol{v}\} \tag{6}$$

$$\boldsymbol{v}_e = [\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_n] \tag{7}$$

式中:m 是滑动窗口大小;f 是非线性激活函数;$\boldsymbol{w}$ 是权重矩阵;b 为偏差项。此外,为了能与后续模块进行计算,增加了一层线性变换 $linear$,使得最终 $\boldsymbol{v}_e = linear(\boldsymbol{v}_e)$。

2.4 Multi-Head Attention

Attention 机制的作用就是对文本中的每个词赋予不同权重,以此让模型更关注于那些权重更高的信息,提升模型学习效率。Multi-Head Attention 能够提供多个经 Attention 计算后的表示子空间,并将多个不同的空间表示做集成,其结构如图 3 所示。

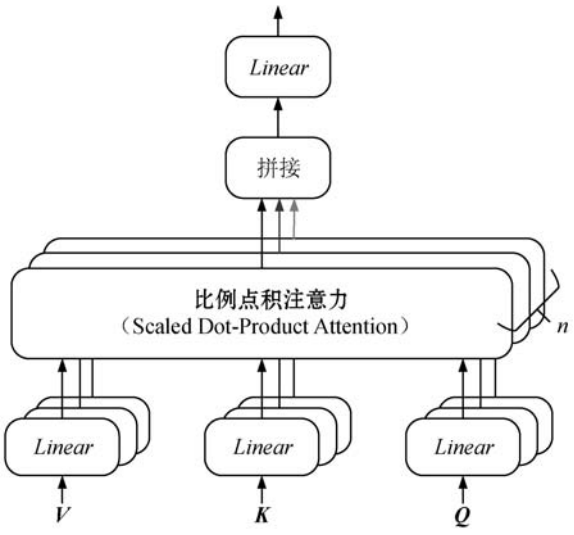

图 3　Multi-Head Attention

Attention 机制通过输入的词向量构建三个不同的向量,分别为 *key* 向量、*query* 向量和 *value* 向量,它们由三个不同的权重矩阵与词向量相乘所得,对于输入 $\tilde{\boldsymbol{e}}$,可以得到整个文本的 $\boldsymbol{K}$ 矩阵、$\boldsymbol{Q}$ 矩阵和 $\boldsymbol{V}$ 矩阵:

$$\begin{cases}\boldsymbol{K}=\tilde{\boldsymbol{e}}\times\boldsymbol{W}^{\mathrm{K}}\\ \boldsymbol{Q}=\tilde{\boldsymbol{e}}*\boldsymbol{W}^{\mathrm{Q}}\\ \boldsymbol{V}=\tilde{\boldsymbol{e}}*\boldsymbol{W}^{\mathrm{V}}\end{cases}\tag{8}$$

然后将 $\boldsymbol{K}$ 矩阵和 $\boldsymbol{Q}$ 矩阵相乘为每个向量计算得分并进行归一化,接着使用 Softmax 激活函数,再与 $\boldsymbol{V}$ 矩阵相乘就得到了包含注意力权重的文本表示 $\boldsymbol{A}$,其计算公式如下:

$$\boldsymbol{A}=\mathrm{Softmax}\left(\frac{\boldsymbol{Q}*\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V}\tag{9}$$

对于 Multi-Head Attention 而言,当有 n 个头时,根据式(9)可得 $\boldsymbol{A}=(\boldsymbol{A}_0,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_{n-1})$ 的集合,将 $\boldsymbol{A}_0$ 到 $\boldsymbol{A}_{n-1}$ 拼接,再与一个权重矩阵相乘得到最终输出 $\boldsymbol{A}_e$:

$$\boldsymbol{A}_e=[\boldsymbol{A}_0,\boldsymbol{A}_1,\cdots,\boldsymbol{A}_{n-1}]*\boldsymbol{W}_e\tag{10}$$

这样,就充分获得了包含全局语义信息的文本表示,每个词的语义信息都得到了加强,更能体现出哪些词语是对标签分类贡献最大的。最后,将 CNN 的输出与 Multi-Head Attention 的输出拼接,得到充分包含全局和局部的语义特征 $\boldsymbol{X}$:

$$\boldsymbol{X}=\boldsymbol{v}_e\oplus\boldsymbol{A}_e\tag{11}$$

2.5　标签注意力

为了更好地结合文本内容考虑标签的相关性,将经过 CNN 处理后的文本表示与标签表示相结合,做交互注意力计算,如图 4 所示。此处仅使用 CNN 输出的本文表示是因为深度学习模型通常伴有过拟合现象的产生,为降低过拟合干扰,不使用更深层次的文本表示,并且 CNN 计算复杂度低,更便于运行。

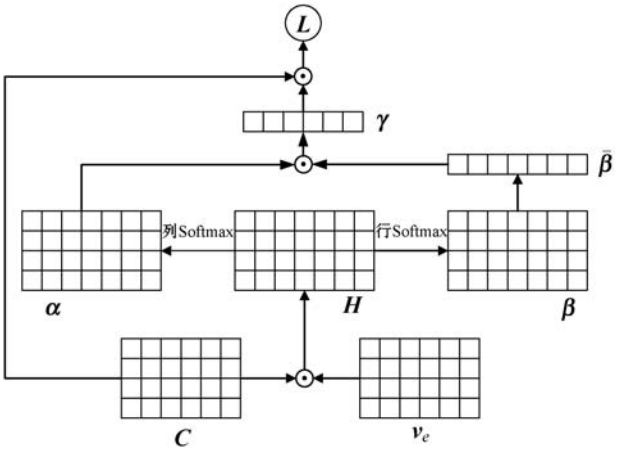

图 4　标签注意力计算

根据图 4(⊙表示点积)可以清楚地看到,首先将标签矩阵与经过 CNN 输出的文本表示 $\boldsymbol{v}_e$ 进行点乘,获取交互矩阵表示 $\boldsymbol{H}$:

$$\boldsymbol{H}=\boldsymbol{C}\odot\boldsymbol{v}_e\tag{12}$$

然后分别进行行与列式的 Softmax 计算,以获取文本对标签和标签对文本的注意力分数 α 和 β:

$$\alpha_{ij}=\frac{\exp(H_{ij})}{\sum_i\exp(H_{ij})}\tag{13}$$

$$\beta_{ij}=\frac{\exp(H_{ij})}{\sum_j\exp(H_{ij})}\tag{14}$$

接着计算 $\bar{\boldsymbol{\beta}}$ 得到文本级的注意力,标签级的注意力则为 $\boldsymbol{\alpha}$ 和 $\bar{\boldsymbol{\beta}}$ 的点积:

$$\bar{\boldsymbol{\beta}}_j=\frac{1}{n}\sum_i\beta_{ij}\tag{15}$$

$$\boldsymbol{\gamma}=\boldsymbol{\alpha}\odot\bar{\boldsymbol{\beta}}^{\mathrm{T}}\tag{16}$$

最终再与标签表示做运算得到基于文本内容的标签表示 $\boldsymbol{L}$:

$$\boldsymbol{L}=\boldsymbol{\gamma}\odot\boldsymbol{C}\tag{17}$$

这样标签表示 $\boldsymbol{L}$ 就非常全面并结合文本内容考虑到了标签与文本的关系以及标签之间的相关性,从而提升了模型的学习能力。

2.6　自适应融合策略

在获得文本语义表示 $\boldsymbol{X}$ 和基于标签的文本表示 $\boldsymbol{L}$ 后,使用一种自适应融合策略将两者结合,目的是为了进一步充分提取有用的语义信息,从而提高模型学习能力,同时也能降低过拟合影响。

本文使用两个权重矩阵 $\boldsymbol{\lambda}_1$ 和 $\boldsymbol{\lambda}_1$ 来确定提取 $\boldsymbol{X}$ 和 $\boldsymbol{L}$ 的信息,两个权重矩阵由 sigmoid 函数获得:

$$\begin{cases}\boldsymbol{\lambda}_1=\mathrm{sigmoid}(\boldsymbol{X}\cdot\boldsymbol{W}_1)\\ \boldsymbol{\lambda}_2=\mathrm{sigmoid}(\boldsymbol{L}\cdot\boldsymbol{W}_2)\end{cases}\tag{18}$$

式中:$\boldsymbol{W}_1$、$\boldsymbol{W}_2$ 为可训练参数矩阵。对于每个标签 i 来说,均有:

$$\lambda_{1i} + \lambda_{2i} = 1 \tag{19}$$

因此,对于最终预测的第 i 个标签的文本表示为:

$$E_i = \lambda_{1i} X_i + \lambda_{2i} L_i \tag{20}$$

2.7 多标签分类

模型最后对 $\boldsymbol{E}$ 通过如下方式进行标签预测:

$$\hat{y} = \text{sigmoid}(\boldsymbol{W} f(\boldsymbol{W}' \boldsymbol{E}^{\mathrm{T}})) \tag{21}$$

式中:$\boldsymbol{W}$ 和 $\boldsymbol{W}'$ 是可训练参数矩阵;f 为 ReLU 激活函数。损失函数使用交叉熵损失:

$$J = -\sum_{i=1}^{n}\sum_{j=1}^{l}(y_{ij}\log(\hat{y}_{ij})) + (1 - y_{ij})\log(1 - \hat{y}_{ij}) \tag{22}$$

3 实验与结果分析

3.1 数据集

本文以两个广泛使用的公开数据集作为基准使用模型,数据集的详细统计信息见表 1。

表 1 数据统计信息

数据集	训练集	测试集	标签数	总词数
AAPD	54 840	1 000	54	69 399
RCV1-V2	23 149	781 265	103	47 236

AAPD:从 arXiv 网站上搜集的论文摘要,包含了 55 840 篇摘要和对应的 54 个标签。

RCV1-V2:由路透社公司提供的 800 000 多篇人工分类的新闻报道组成,每篇新闻有多个标签,共有 103 个标签。

3.2 基线模型

本文将选择以下几种主流且较新的模型作为基线模型进行对比:

(1) XML-CNN[8]:结合 CNN 在文本分类中的优势,设计了一个动态池化层来适应多标签的文本分类。

(2) EXAM[12]:利用交互机制来计算词与标签的匹配得分,然后将这些分数聚合成每个类的预测。

(3) SGM[13]:模型将 MLTC 任务视为一个序列生成问题,将标签相关性考虑在内,在编码器阶段使用双向 LSTM 计算每个单词的隐层表示,然后使用 Attention 机制让模型关注于对标签贡献更大的词,最后使用 LSTM 进行解码后进行分类。

(4) AttentionXML[14]:使用双向 LSTM 来捕获单词的长距离依赖关系,使用多标签 Attention 机制捕获文本中与每个标签最相关的部分,构建具有标签信息的文本表示来进行分类。

上述基线模型均使用其文献中所提供的源代码进行实验,为体现公平性和鲁棒性,参数均按各文献原文所属保持不变,同时各模型的数据集预处理方式保持一致。

3.3 实验设置

(1) 本文实验平台的相关配置如表 2 所示。

表 2 实验平台

实验环境	具体信息
操作系统	Ubuntu16.04
GPU	NVIDIA GTX 1080Ti
内存	32 GB
开发语言	Python 3.6
开发框架	Torch 1.1.0
开发工具	PyCharm

(2) 模型参数设置:batch size 为 32,使用 Adam 更新参数,学习率为 0.001,词向量维度 300,dropout 为 0.4,卷积核窗口(2,3,4),卷积核数量 100,多头数量为 10。

(3) 评估指标:使用精度(Precision at K,P@K)和归一化折损累计增益(Normalized Discounted Cumulative Gain at K,NDCG@K)来进行评估,其公式如式(23) - 式(25)所示。

$$P@K = \frac{1}{k}\sum_{l \in rank_k(\hat{y})}^{k} y_l \tag{23}$$

$$G_{\mathrm{DC}}@K = \sum_{l \in rank_k(\hat{y})} \frac{y_l}{\log(l+1)} \tag{24}$$

$$G_{\mathrm{NDC}}@K = \frac{DCG@K}{\sum_{l=1}^{\min(k, \| y \|_0)} \frac{1}{\log(l+1)}} \tag{25}$$

式中:$rank_k(\hat{y})$ 表示真实标签在预测标签的前 k 个索引;$\| \boldsymbol{y} \|_0$ 为在真实标签向量 $\boldsymbol{y}$ 中相关标签的个数。

3.4 实验结果与分析

表 3 和表 4 分别展示了本文模型与基线模型在两个数据集上的表现情况。

表 3 在 P@K 上的结果对比(%)

数据集	评估指标	XML-CNN	EXAM	SGM	AttentionXML	Our Method
AAPD	P@1	74.37	83.26	75.67	83.02	**84.27**
	P@3	53.84	59.77	56.75	58.72	**60.32**
	P@5	37.79	40.66	35.65	40.56	**41.11**
RCV1-V2	P@1	95.75	93.67	95.37	96.41	**96.62**
	P@3	78.63	75.80	81.36	80.91	**81.77**
	P@5	54.94	52.73	53.06	56.38	**56.90**

表 4　在 G_{NDC}@K 上的结果对比(%)

数据集	评估指标	XML-CNN	EXAM	SGM	AttentionXML	Our Method
AAPD	nDCG@1	74.37	83.26	75.67	83.02	**84.27**
	nDCG@3	71.12	79.10	72.36	78.01	**79.93**
	nDCG@5	75.93	82.79	75.35	82.31	**83.85**
RCV1-V2	nDCG@1	95.75	93.67	95.37	96.41	**96.62**
	nDCG@3	89.89	86.85	91.76	91.88	**92.60**
	nDCG@5	90.77	87.71	90.69	92.70	**93.13**

从表 3 和表 4 的实验结果对比中可以清楚地看到,本文模型在 P@K 和 G_{NDC}@K 上相比基线模型都取得了最优结果。在 AAPD 数据集上,XML-CNN 在两种指标上的整体性能最差,原因是其只考虑了文本语义信息,没有将标签相关性考虑进去,也没有考虑不同单词的贡献程度,使得模型学习能力大大受限。EXAM 则在基线模型中表现最佳,它通过一个交互机制计算词与标签的匹配得分,考虑到了标签与文本间的相互影响,但同 SGM 和 AttentionXML 模型一样,都没有更深层次地学习标签之间的相关性以及不同标签对文本的表示,因此较于本文模型均较差。在 RCV1-V2 数据集上,所有基线模型与本文模型都表现不错,但本文模型相比基线模型在两个评估指标上仍均有提升,且都获得了最优结果。基线模型中在整体性能上表现最差的为 EXAM,而表现最好为 AttentionXML,主要原因是 EXAM 更注重标签与文本的语义关联,而 RCV1-V2 数据集的类别较为明确,且总词数较少,对挖掘深层次的文本语义与标签关联信息极易造成过拟合现象,从而导致在测试集上分类精度的降低,AttentionXML、XML-CNN 模型则相对更注重文本语义理解上,在该数据集上学习效率会更高。

纵观两个数据集的整体结果,各基线模型在数据集上的表现有好有差,而本文模型均能取得最优效果,可见本文模型具有较好的鲁棒性。对于不同数据集来说,模型的适应能力非常重要,本文所提模型中使用的自适应融合策略就能非常好地适应不同数据集,对类别较为明确、标签间关联程度较小的 RCV1-V2 数据集,能够自动降低 $\boldsymbol{L}$ 的权重,对 AAPD 数据集来说则会适当提升权重。在训练速度和计算复杂度上,CNN 计算速度更快,且能避免梯度爆炸和消失的情况,因而本文模型与基线模型保持了相当水平,算力要求适中,结合评估结果来看,本文模型在训练效率上具有优势。

此外,为进一步验证本文所提模型每一个模块的作用和有效性,以 AAPD 数据集为例进行消融实验,分别研究在 P@K 和 G_{NDC}@K 上的结果。在本文所提模型基础上,分别去除 CNN 模块(记为 w/o C)、多头注意力机制模块(记为 w/o M)和标签注意力模块(记为 w/o L)。模型消融实验的结果如图 5 所示。

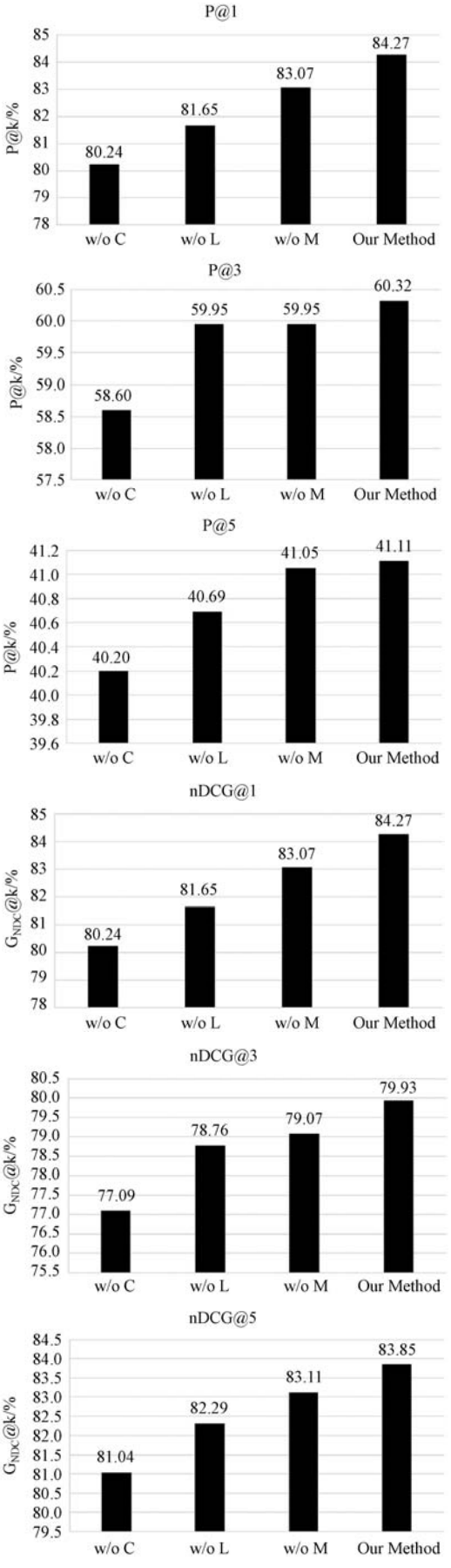

图 5　消融实验结果对比

可以清晰地看到,无论去掉哪个模块,均一定程度降低了模型精度,初步表明本文所提模型是合理且有效的。分模块来看,去掉 CNN 的模型表现在两个指标上都是最差的,表明 CNN 对文本语义的提取极为关键。事实上,文本所含的各个标签通常只对文本中某一部分更为关注,CNN 恰好能够很好地提取本文的局部信息,因此非常符合这种情况。若去掉多头注意力机制对模型的影响最小,多头注意力更注重文本全局的语义信息,这对于多标签任务而言不能很好地学习到标签所对应文本内容间的差异性,因此对模型的整体性能提升较小,但它也还是能为模型对文本语义的理解带来帮助。再结合 XML-CNN 的结果来看,本文模型所引入的标签注意力表示,能够极大提高模型性能,印证了标签注意力模块的有效性;若去掉标签注意力模块也同样降低了模型性能,可见结合文本内容后考虑标签的相互关系确实有助于提高分类精度。标签信息和文本信息都是相互依赖的,标签的相互关系能够通过文本内容所体现,因此单单考虑标签间相关性不足以提升模型学习能力。

综上所述,本文模型使用 CNN 和多头注意力机制充分挖掘了文本的局部和全局语义信息,并使用标签注意力结合文本内容全面考虑了标签间的相互关系,最后使用自适应融合策略进一步提取所需语义信息,提升了分类性能,实验结果表明了所提模型的合理性和有效性。

4 结 语

本文提出了一种基于注意力和 CNN 的多标签文本分类模型,不仅充分挖掘了文本的语义信息,还很好地考虑了结合文本内容后的标签之间的相关性,有效解决了前人模型的缺点。在与当前最新且主流的基线模型对比中,获得了优异的分类效果,体现出很好的鲁棒性,具有很高的效率。在消融实验中,也进一步验证了本文所提模型的合理性和有效性,模型能够通过 CNN、多头注意力机制和标签注意力提高模型学习能力,取得较为理想的效果。

由于文本数量的激增,人工标注耗时耗力,因此在未来工作中,将进一步研究在小样本学习中多标签分类的问题,以此适应当今分类需求,同时降低计算复杂度、提高训练速度、优化模型算法也将是接下来的重点。

参 考 文 献

[1] 谢先章,王兆凯,李亚星,等. 基于卷积神经网络的跨领域语义信息检索研究[J]. 计算机应用与软件,2018,35(8):73 - 78.

[2] 徐菲菲,芦霄鹏. 结合卷积神经网络和最小门控单元注意力的文本情感分[J]. 计算机应用与软件,2020,37(9):75 - 80,125.

[3] 傅健. 卷积深度神经网络再基于文档的自动问答任务中的应用与改进[J]. 计算机应用与软件,2019,36(8):177 - 180,219.

[4] 吴迪. 基于信息融合的文本主题分析算法研究[D]. 成都:电子科技大学,2020.

[5] Kim Y. Convolutional neural networks for sentence classification[C]//Conference on Empirical Methods in Natural Language Processing,2014:1746 - 1751.

[6] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Conference on Empirical Method in Natural Language Processing,2014:1532 - 1543.

[7] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[EB]. arxiv:1607.01759,2016.

[8] Liu J Z, Chang W C, Wu Y X, et al. Deep learning for extreme multi-label text classification[C]//40th International ACM SIGIR Conference on Research,2017:115 - 124.

[9] Chen G B, Ye D H, Xing Z C, et al. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization[C]//International Joint Conference on Neural Networks,2017:2377 - 2383.

[10] Kurata G, Xiang B, Zhou B. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence[C]//Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2016:521 - 526.

[11] Zhang W J, Yan J C, Wang X F, et al. Deep extreme multi-label learning[C]//ACM on International Conference on Multimedia Retrieval,2018:100 - 107.

[12] Du C X, Chen Z, Feng F L, et al. Explicit interaction model towards text classification[C]//AAAI Conference on Artificial Intelligence,2019:6359 - 6366.

[13] Yang P C, Sun X, Li W, et al. SGM: Sequence generation model for multi-label classification[EB]. arXiv:1806.04822,2018.

[14] You R, Dai S, Zhang Z, et al. AttentionXML: Extreme multi-label text classification with multi-label attention based recurrent neural networks[EB]. arXiv:1811.01727,2018.

[15] 肖琳,陈博理,黄鑫,等. 基于标签语义注意力的多标签文本分类[J]. 软件学报,2020,31(4):1079 - 1089.

[16] 王浩镔,胡平. 采用多级特征的多标签长文本分类算法[J]. 计算机工程与应用,2021,57(15):193 - 199.

[17] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[EB]. arXiv:1301.3781,2013.