2020 年 10 月 第 10 期
Vol. 41 No. 10 2020

基于配对排序损失的文本多标签学习算法

顾天飞,彭敦陆

(上海理工大学 光电信息与计算机工程学院,上海 200093)
E-mail: pengdl@ usst. edu. cn

摘 要: 文本多标签学习是一项重要的自然语言处理任务,是对信息进行有效管理的一项关键技术. 该任务需同时考虑到对文本和多标签进行建模. 基于此,论文首先利用预训练语言模型 BERT 对文本序列进行特征提取,然后采用配对排序损失作为模型训练的目标函数,以对多标签之间的排序关系进行建模. 最后为得到更精准的预测结果,加入了辅助的标签阈值学习. 考虑到基于铰链函数的排序损失存在训练困难的现象,为此论文采用了一种光滑的替代损失,并从理论上验证了其有效性. 在真实文本数据集上的实验表明,本文提出的算法能提供更好的性能从多标签分类和排序两方面.
关 键 词: 深度学习; 多标签学习; 文本分类; 配对排序损失
中图分类号: TP391 文献标识码: A 文 章 编 号: 1000-1220(2020)10-2045-06

Multilabel Text Learning Algorithm Based on Pairwise Ranking Loss

GU Tian-fei, PENG Dun-lu
(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Multilabel text learning, as an important natural language processing task, is a key technology for information management. This task needs to take into account the modeling of text and multilabel simultaneously. In this paper, we first apply a pre-trained language model BERT for feature extraction of text, and then use pairwise ranking loss as the optimization function so as to model the ranking relationship between labels. Finally, an auxiliary threshold prediction model is proposed to obtain the accurate predictions. The hinge-based loss is difficult to optimize especially with deep networks. We propose a smooth surrogate pairwise ranking loss as the boundary of hinge loss and theoretically verifies its effectiveness. Experiments conducted on the public Chinese text dataset demonstrate the effectiveness of proposed algorithm from the perspective of classification and ranking.
Key words: deep learning; multilabel learning; textclassification; pairwise ranking loss

1 引 言

文本分类是自然语言处理领域中的一项重要任务,是构建信息检索、对话机器人等复杂系统的基础. 多分类假设类别之间是互斥的,即一篇文档有且只能归属于单个类别. 而事实上,对象是多语义的,比如一篇新闻能同时标注上“体育”和“足球”标签. 所以,多标签更适合用来对现实问题进行建模,并有其实际的应用背景和学术价值.

多标签学习存在多标签分类和标签排序两类任务[11],前者将标签集划分为与样本相关和不相关两部分,后者则预测标签之间的前后关系. 上述两项任务存在共通性,多标签分类和标签排序之间是可以相互转换的,文献引入校准标签对排序的标签进行划分[14],而采用判别模型完成多标签分类时,样本对标签的后验概率天然具有可排序性[5]. 故而,学界和业界开始尝试将两项任务联合起来进行解决,并运用于不同的应用领域[6,9,16]. 大体上,这类方法基于以下思想,得分较高的标签更能体现样本的语义,模型应使正标签集排在负标签集之前,这样筛选出来的标签也更加精准[16],从这一角度看,标签排序考虑到了标签的相对关系.

对于文本处理,过去的研究普遍采用文本特征手工提取的方式[19,20]. 得益于深度学习的发展,端到端的深度表征模型已成为当今的主流[1-5,7,8,15]. 与此同时,深度模型的性能受到标注数据缺失和语义提取不足的限制. 为此,本研究引入迁移学习,将 BERT[1] 作为模型的特征提取部分,将多标签分类和排序共同纳入考虑,利用标签之间的相对关系来增强多标签预测的有效性. 文献普遍采用错误排序统计[10] 和铰链损失[9] 刻画多标签排序误差,但这些损失函数通常难以优化,尤其在深度模型的背景下. 故本文采用一种替代的配对排序损失,该损失函数在实数域上可微,同时也是铰链损失的边界. 此外,为了更准确地获得文本实例对应的标签集,标签的筛选被看作为一项二值分类,用一个辅助网络构建筛选标签的阈值.

本文的贡献如下: 1) 将迁移模型 BERT 运用于文本多标签学习; 2) 提出配对排序目标函数对标签排序任务进行建模,并给出了相应的理论分析. 最后,为了决断出精准的标签集,算法引入额外的辅助网络进行阈值预测.

收稿日期: 2019-12-17 收修改稿日期: 2020-01-21 基金项目: 国家自然科学基金项目(61772342,61703278)资助. 作者简介: 顾天飞,男,1994 年生,硕士研究生,研究方向为深度学习、机器学习; 彭敦陆,男,1974 年生,博士,教授,CCF 会员,研究方向为大数据管理、Web 数据管理、自然语言处理、深度学习.

2 相关工作

一般地,解决多标签任务存在两类思路,问题转换和算法适应[11].前者将多标签学习转化为二值分类[11]、多分类[12]或标签排序[14],后者则修改现有的学习算法以适应多标签领域[13].上述技术主要集中于传统机器学习,往往存在严重的性能瓶颈,计算规模和标签空间呈正比[10-14].如今,神经网络在模式识别领域获得了巨大的成功,其中很大一部分运用到了多标签学习中[3,5-9].

传统的文本分类算法受限于语义和句法信息提取能力的不足,深度模型已经成为了该领域的主流.文献[15]率先采用词向量 word2vec[17]进行词嵌入和卷积神经网络作为特征提取器,获得了显著的性能提升.该模型奠定了深度文本分类的一种范式,即模型一般由词嵌入层、衔接模型和分类器三部分组成.如何通过海量的语料库无监督学习到词的表示是一项热门研究,Word2vec[17]通过对词语上下文和语义关系进行建模,将词语嵌入到稠密的欧式空间中.BERT[1]由多层 Transformer[4]构建而成,能解析出更深层的语义,并能适用于各项下游任务.

文本多标签学习需要考虑到两方面,文本信息的提取和标签之间的相关性,现有的研究基本上是围绕这两方面展开的.一部分研究构建了基于卷积神经网络的模型[5,7,8],文献[5]采用了二值交叉损失对多标签进行建模,文献[8]引入指示神经元对标签共现进行建模,以利用标签的信息,文献[5]将标签预测看作为序列生成,引入循环神经网络构建标签之间的关系.文本序列的各个位置对标签的影响是不同的,SGM[5]利用注意力机制加强模型的关注性.

排序学习的目的是通过机器学习算法对项目进行排序,在信息检索、推荐系统中运用极为广泛.多标签学习存在以下假设,与样本相关的标签在排序上高于不相关的标签,所以排序任务能很好的刻画这种标签关系.文献[16]最早将文本多标签分类看作为一项排序任务,并利用配对排序损失刻画误差,但文献仅在多层感知机模型上验证了损失函数的有效性.配对排序损失也可以应用于图像检测领域[6,9],但研究中普遍采用的铰链损失存在训练困难的问题.为了弥补上述缺点,本研究在深度文本多标签学习背景下,尝试了语言模型的迁移学习,并着重了探讨了配对损失的使用.

3 本文工作

本章将首先给出问题的定义,然后提出结合 BERT 的文本特征提取模型,最后给出配对排序和标签阈值预测的设计,以及相关的目标函数.

3.1 问题描述

给定多标签数据集 $D=\{(x_i,Y_i)\}_{i=1}^{N}$,其中 $x_i=w_1,w_2,\cdots,w_l$ 为文本输入序列,$Y_i\subseteq y$ 为与样本相对应的标签集,$y=\{\lambda_1,\lambda_2,\cdots,\lambda_L\}$ 为任务指定的标签域.用 y_i 向量化 Y_i,$y_{i,k}=1$ 表示第 k 个标签属于样本 x_i.本研究的目的是从 D 中学习到一项多标签文本分类器 $F(x)$,对于样本 x,该分类器不仅可以得到样本与各标签之间的相关性得分 $s=s(x):X\rightarrow\mathbb{R}^L$,也可以根据 s 得到最终的预测结果 $\hat{Y}=g(s)$,g 为决断函数.$F:g(s)$ 构成了多标签分类器.

定义 1. 多标签排序任务,给定样本 x,若 s^* 为理想的映射函数,则需满足以下性质:

$$s_m^*>s_n^*,\forall m\in Y,n\in\bar{Y}=Y-Y \tag{1}$$

这里,s_m^*,s_n^* 分别表示得分 s^* 的第 m,n 分量,$\bar{Y}$ 为 Y 的补集.定义 1 对 s 进行了约束,与实例相关的标签应具有更高的得分,换言之,$s_n\geqslant s_m$ 是不理想的情况,在预测中需避免.

3.2 多标签文本学习模型

在深度自然语言处理中,一个端到端模型一般由以下几个步骤组成,首先将原始文本序列嵌入至稠密的表征词嵌入 $h_1,h_2,\cdots,h_l$,其次通过衔接模型将词嵌入序列转化为定长的表征向量,最终输入到文本分类器中.对词嵌入表征的研究和应用向来受到学界和业界的广泛关注,通过预训练词向量使词嵌入涵盖语义和语法信息.然而,类似于 Word2vec 词向量模型存在无法解析一词多义,上下文信息缺失等缺点,往往对性能的提升并不明显.BERT 作为一种语言迁移模型,可以较好地弥补上述缺陷.

在词嵌入阶段,$bert(\cdot)$ 将原始文本序列 x 中的每个元素映射到固定尺寸的嵌入,映射方式如下:

$$h_1,h_2,\cdots,h_l=bert(w_1,w_2,\cdots,w_l) \tag{2}$$

这里,$h\in\mathbb{R}^d$,$d>L$ 的维度由 $bert(\cdot)$ 决定.衔接模型用于对嵌入进行整合,文献中,通常会垒砌大量模型[5,15],对于这一环节本研究不做过多地复杂化,采用均值操作 $mean(\cdot)$ 将嵌入序列转化为 d 维的特征向量 f:

$$f=mean(h)=\frac{\sum_i h_i}{l} \tag{3}$$

接下来,考虑标签相关性得分的建模,由 d 维特征向量向 L 维向量映射,形式化为:

$$s=relu(W_s f+b_s) \tag{4}$$

其中,$W_s\in\mathbb{R}^{L\times d}$ 为权重矩阵,$b_s\in\mathbb{R}^L$ 为偏置向量.式(4)中的 $relu(\cdot)$ 为神经网络的激活函数.至此,对某个输入样本 x,便能得到模型对各个类别的打分 s,即为类别对样本的相关性.多标签和多类别分类在判决函数上存在一定差异.多类别假设类别之间是相互独立的,故而往往取得分最大的类别作为输出标签.在多标签分类中,每个实例对应的标签数是不同的.简单的做法是取前 k 最大得分或设置全局阈值(将得分大于某一阈值的标签筛选出来),这些方法会造成额外的预测误差.本研究将采取一种更灵活的做法,即让 $g(\cdot)$ 作为一项可学习的函数,为每个标签自动地学习得到适应于样本特征 f 的阈值.阈值建模类似于标签相关性得分模型:

$$\theta=relu(W_{thr}f+b_{thr}) \tag{5}$$

模型的预测同时依赖于式(4)和式(5):

$$\hat{Y}_i=\{k\mid s_{i,k}\geqslant\theta_{i,k},k=1,\cdots,L\} \tag{6}$$

上式中,$s_{i,k}$ 表示样本与标签的相关性得分 s_i 的第 k 分量,$\theta_{i,k}$ 表示阈值的第 k 分量.图 1 为模型的整体框架.

3.3 多标签配对排序损失

上节介绍了结合语言迁移模型的多标签分类模型,本节将引出如何对模型参数进行优化.形式上,需要解决如下优化问题:

$$L_s=\frac{1}{N}\sum_{i=1}^{N}l(s(x_i),Y_i)+R(\Phi_s) \tag{7}$$

这里，l 为每个样本上的损失项，R 为模型参数的正则项，$\Phi_s = [\![W_s, b_s]\!]$ 为标签相关性得分模型的参数. 在训练式(7)

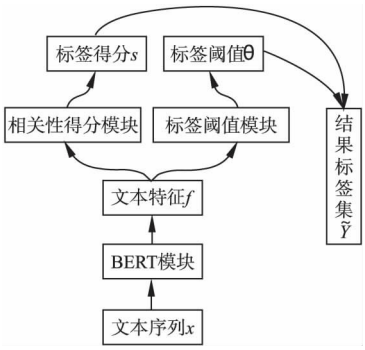

图 1　算法框架

Fig. 1　Architecture of algorithm

时，解冻 $bert$，对其进行参数微调. 由定义 1 可得，属于 Y 的标签得分需尽可能地大，反之亦然. 借鉴三元损失，易对损失进行建模:

$$l_s = \frac{1}{|Y||\bar{Y}|}\sum_{n\in Y}\sum_{m\in\bar{Y}}\max(0, \alpha + s_n - s_m) \tag{8}$$

式(8)采用了铰链损失，α 是一项超参数，用来设定相关与不相关标签之间的边界. 该损失函数是非光滑的，在 $x=0$ 处不可微，从而造成了优化的困难. 为解决上述问题，本研究考虑引入替代损失:

$$l_s = \frac{1}{|Y||\bar{Y}|}\sum_{n\in Y}\sum_{m\in\bar{Y}}\ln(1+\beta e^{s_n - s_m}) \tag{9}$$

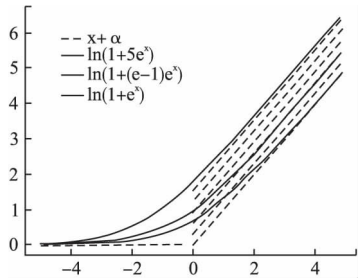

图 2　损失函数 l_s 的性质

Fig. 2　Property of loss function l_s

上式中，β 是常系数. 替代损失式(9)是式(8)光滑的近似. 由图 2 中实线可见，该损失函数为实数域 $\mathbb{R}$ 上处处可微的凸函数，在 $\mathbb{R}^+$ 上为铰链损失的边界，当且仅当，$\beta = e^{\alpha} - 1$. 此外，β 值越小，则实线越接近 y = 0. 章节 4 给出了相应的梯度求解，并且从经验误差最小化和贝叶斯最优预测角度进行理论分析.

在多标签领域，配对排序损失的计算复杂度与标签域的大小有关 $O(|Y||\bar{Y}|)$，当标签数量 L 过多时，会产生大量的配对. 实践中，一种有效的改善措施是对所有配对进行子集采样 $p_t(Y) \in Y\otimes\bar{Y}$，$\otimes$ 为笛卡尔积，t 为子采样的数量. 于是式(9)可以写成以下形式:

$$l_s = \frac{1}{t}\sum_{(m,n)\in p_t(Y)}\ln(1+\beta e^{s_n - s_m}) \tag{10}$$

3.4　阈值模型损失

式(5)为阈值回归模型，根据样本特征为每个标签学习筛选阈值 θ，并通过式(6)得到最终的预测标签集. 对某个标签来说，预测可以转换为一项二值问题，得分大于阈值为正样本，反之作为负样本. 于是，阈值参数的目标函数可以写成以下形式:

$$l_{thr} = -\frac{1}{L}\sum_{k=1}^{L} y_{i,k}\log(sigmoid(s_{i,k} - \theta_{i,k})) \tag{11}$$

这里，l_{thr} 为交叉熵，$sigmoid = \frac{1}{1+e^{-x}}$ 将判别结果约束至概率空间中. 训练过程中，冻结 $bert$ 的参数和 Φ_s，对阈值模型的参数 $\Phi_{thr} = [\![W_{thr}, b_{thr}]\!]$ 进行优化.

4　理论分析

本章首先对优化目标函数进行梯度计算，考虑式(10)对 $\boldsymbol{s}_m$ 和 $\boldsymbol{s}_n$ 的梯度为:

$$\frac{\partial l_s}{\partial \boldsymbol{s}_n} = \sum_{(m,n)\in p_t(x)}\frac{\beta e^{s_n - s_m}}{1+\beta e^{s_n - s_m}} \tag{12}$$

$$\frac{\partial l_s}{\partial \boldsymbol{s}_m} = \sum_{(m,n)\in p_t(x)}\frac{-\beta e^{s_n - s_m}}{1+\beta e^{s_n - s_m}} \tag{13}$$

整合式(12)和式(13)可以得到:

$$\frac{\partial l_s}{\partial \boldsymbol{s}} = \sum_{(m,n)\in p_t(x)}\frac{\beta\xi_{n,m}e^{s^T\xi_{n,m}}}{1+\beta e^{s^T\xi_{n,m}}} \tag{14}$$

这里，$\xi_{n,m}$ 为 L 维向量，其中第 n 项为 +1，第 m 项为 −1，其余项为 0，以上计算结果说明说明了目标函数在实数域上式处处可微的. 文献[18]从经验误差最小化和贝叶斯最优预测角度，证明了排序统计的有效性. 相同地，对损失函数式(9)进行理论分析，式(10)作为简化版本同理可得. 考虑贝叶斯预测准则:

$$s_k(x) = p(k\in Y|x) = \sum_{Y\in y, k\in Y} p(Y|x) \tag{15}$$

上式决定了标签 λ_k 的得分即相应的排序，$p(k\in Y|x)$ 为标签域中所有可能的标签集的边际分布.

定理 1. 采用损失函数式(9)能达到经验损失最小化.

证明: 考虑损失函数经验误差最小化:

$$R(s) = \mathbb{E}[l_s(s(x), Y)] \tag{16}$$

将式(16)改写成条件经验损失的形式:

$$\begin{aligned} R(s|x) &= \mathbb{E}[l_s(s(x), Y)|x] \\ &= \sum_{Y\in y} p(Y|x)\, l_s(s(x), Y) \\ &= \sum_{Y\in y} p(Y|x) \sum_{(m,n)\in Y\otimes\bar{Y}} \gamma_{m,n} \\ &= \sum_{m,n} p((m,n)\in Y\otimes\bar{Y}|x)\,\gamma_{m,n} \end{aligned} \tag{17}$$

这里，$\gamma_{m,n} = \ln(1+e^{s^T\xi_{m,n}})$. 现需找到使经验损失最小化的得分函数 s^*，即尽可能满足定义 1. 计算式(17)的一阶和二阶导:

$$\frac{\partial R(s|x)}{\partial \boldsymbol{s}} = \sum_{m,n}\delta_{m,n}\frac{\xi_{n,m}^T e^{s^T\xi_{n,m}}}{1+e^{s^T\xi_{n,m}}} \tag{18}$$

$$\frac{\partial^2 R(s|x)}{\partial \boldsymbol{s}^2} = \sum_{m,n}\delta_{m,n}\frac{\xi_{n,m}\xi_{n,m}^T e^{s^T\xi_{n,m}}}{(1+e^{s^T\xi_{n,m}})^2} \tag{19}$$

这里，$\delta_{m,n} = p((m,n)\in Y\otimes\bar{Y}|x)$. 分析式(19)，$\xi_{n,m}\xi_{n,m}^T$ 是半正定的，故能保证在实数域上 $R(s|x) > 0$，即经验损失是凸函数，且必存在最优的 s^*. 现通过 $\frac{\partial R(s|x)}{\partial s} = 0$ 得到最优函

数,简得:

$$g(m,n)=\sum_{m,n}\delta_{m,n}e^{s^T\xi_{n,m}}=0 \tag{20}$$

$$lng(m,n)=\sum_{m,n}(ln\delta_{m,n}-s^T\xi_{m,n}) \tag{21}$$

替换式(21)中的(n,m),得到:

$$lng(n,m)=\sum_{m,n}(ln\delta_{n,m}+s^T\xi_{m,n}) \tag{22}$$

$$s^T\xi_{m,n}=ln\frac{\delta_{m,n}}{\delta_{n,m}} \tag{23}$$

根据式(23),对于最优的标签得分函数s^*,根据定义1,需满足$s_m^*>s_n^*$,当且仅当满足$\delta_{m,n}>\delta_{n,m}$:

$$p((m,n)\in Y\otimes\bar{Y}|x)>p((n,m)\in Y\otimes\bar{Y}|x) \tag{24}$$

式(24)易得$p(m\in Y|x)>p(n\in Y|x)$,基本满足贝叶斯预测准则. 综上,以式(9)作为排序损失,能达到经验损失最小化.

5 实验评估

本章节将在真实的中文文本数据集上验证本文所提算法的性能,实验首先对比了不同的标签决断方法和损失函数的表现,最后与一些主流的方法进行比较.

5.1 实验数据

本实验选用了法研杯比赛CAIL2018[1]罪名预测任务,来进行算法验证. 为减少训练时间,选取了187100份样本,并根据8:1:1的比例将数据集划分为训练集,测试集和验证集. 多标签数据集存在额外的性质,表1给出相关的信息. 在文献中,Card和Dens分别表示样本所属标签平均数量和标签密度. 标签集数量较大说明存在大量标签共现的情况,如何利用上标签的关系显得额外重要.

表1 多标签信息

Table 1 Data set information

任务	标签	标签集	Card	Dens
罪名预测	202	3541	1.214	0.006

5.2 实验设置

1) 实验平台: 本研究中所有的代码都由Python编写,模型基于Tensorflow搭建. 采用哈工大提供的BERT[2]预训练模型,该版本在海量的中文语料库上完成训练,并在各项中文任务验证了其有效性. 设备系统为Ubuntu16.04,配备两块NVIDIA GeForce 1080Ti显卡,内存为64G.

2) 数据预处理: 原始文本数据已经做了脱敏处理,本实验将作进一步地优化,去除了文档中的特殊符号,西文字符等. 由于文书是存在格式的,其中有些子句实际上是无用的,比如"人民检察院指控","公诉机关指控"或者文书审理日期等,实验中将上述字符串从文档中剔除. 为处理数据集存在的多标签不平衡问题,这里首先按照50:50的比例将标签集划分为多数类和少数类,并对少数类进行上采样处理.

3) 实验参数设置: 第一阶段对标签得分模型进行优化,该阶段解冻*bert*的参数,做参数微调. 第二阶段冻结*bert*,仅对阈值模型进行优化. 两个阶段皆采用ADAM优化器,学习率设置为0.001. BERT模型输入序列的尺寸上存在限制,最大输入为512,训练中将长文本按200字符为单位进行分割,模型预测过程中,将由各个划分的特征均值作为完整文本的特征. 式(10)配对子采样的数量为120. 由式(14)可知,超参数β是一项平滑参数,对梯度的尺度和训练的收敛性存在一定影响,与学习率的功能是相似的. β过大会使损失函数趋向于线性,过小则趋向于为零,在超参数调优过程中,尝试了区间0.1至2都能使训练收敛,故方便起见这里设置为1.

4) 评价指标: 本研究同时考虑到了多标签的分类和排序两方面,所以实验也将从这两方面对预测结果进行评估. 下面所阐述的评价指标都参考自文献[11],采用宏观和微观F_1得分衡量分类性能:

$$MacroF_1=\frac{1}{L}\sum_k eval(TP_k,FP_k,TN_k,FN_k) \tag{25}$$

$$MicroF_1=eval(\sum_k TP_k,\sum_k FP_k,\sum_k TN_k,\sum_k FN_k) \tag{26}$$

这里,$eval=2\cdot prec\cdot recall/(prec+recall)$为$F_1$得分,用于调和准确率$prec=TP/(TP+FP)$,召回率$recall=TP/(TP+FN)$. 在以上式子中,*TP*表示为真正样本,*FP*为假正样本,*TN*为真负样本,*FN*为假负样本. 用排序误差衡量排序性能:

$$RankL=\frac{1}{N}\sum_i\frac{1}{|Y_i||\bar{Y}_i|}\left|\left\{(y_{i,m},y_{i,n})\left|\begin{matrix}s_{i,m}<s_{i,n},\\(y_{i,m},y_{i,n})\in Y_i\otimes\bar{Y}_i\end{matrix}\right.\right\}\right| \tag{27}$$

排序损失*RankL*统计预测结果中的对误排标签对,数值越小越好.

5.3 实验结果分析

本章节将通过实验评估本文所提算法的有效性.

实验1. 不同标签决断方法

在章节3.2中提到了其它两种标签决断方法,Top-k和全局阈值,在使用中,k值取1、3和5,阈值从0.05-0.95按照0.01为间隔,表2展示测试集上最优得分. 值得注意的是,本文提出的得分模型其输出是映射到实数域上的,所以通过*sigmoid*将其约束到概率空间中. 相对来说,宏观和微观指标衡量了模型的整体分类性能,对标签的误选较为敏感,Top-k和全局阈值是静态的刷选策略,而没有考虑到了样本特征本身所携带的信息,从而造成得分上的下滑. 并且,在使用这些算法的时候往往会遇到超参的优化问题. 表2中的结果说明在多标签领域,标签决断对最终预测结果的影响非常大. 相比于全局阈值,阈值预测方法在分类指标上能提供2%的提升,排序指标上也是表现最优的.

表2 标签决断技术的对比

Table 2 Comparison of label decision

算法	$MacroF_1\uparrow$	$MicroF_1\uparrow$	$RankL\downarrow$
Top-k	0.523	0.735	0.074
全局阈值	0.712	0.824	0.069
阈值预测	0.738	0.846	0.048

实验2. 不同训练方式

[1] https://github.com/thunlp/CAIL

[2] https://github.com/ymcui/Chinese-BERT-wwm

本实验将配对排序损失和其它几种目标函数进行比较:

1) 二值交叉损失[5] (BCE):

$$l_{bce}=-\frac{1}{L}\sum_{k=1}^{L}y_{i,k}\log(sigmoid(s_{i,k})) \tag{28}$$

BCE 相当于标签转换,类似于参数共享的二值分类模型;

2) 铰链损失式(8);

3) BP-MLL[8] 基于指数损失. 为了确保方法之间的可对比性,实验采用同一套数据预处理技术,并且默认采用阈值预测技术. 表 3 展示了各种训练方式之间的性能对比. 可以看到 BCE 在微观指标上的表现略微占有,但在其余指标上,文本的算法存在竞争性的优势. 这是由于 BCE 注重整体的分类误差,配对排序损失则考虑错误的排序对. 宏观指标是标签 F_1 得分的平均,本文的算法在 $MacroF_1$ 上的优势也体现了数据不平衡对配对排序损失的影响较低.

表 3　训练方式之间的对比

Table 3　Comparison of training approaches

算法	$MacroF_1\uparrow$	$MicroF_1\uparrow$	$RankL\downarrow$
BCE	0.701	0.850	0.063
铰链损失	0.450	0.671	0.145
BP-MLL	0.722	0.818	0.059
本文的算法	0.738	0.846	0.048

实验 3. 不同模型进行对比

前两项实验分别从标签决断和训练方式做了对比,本实验将选取一些常用的多标签算法进行完整的对比:

1) 二值相关 BR[11] 为每个标签训练一个 SVM 分类器;

2) ML-KNN[13] 将 KNN 拓展到多标签领域,是一种惰性学习器;

3) 卷积神经网络 CNN[5] 是最常用的深度文本模型;

4) CNN-RNN[7] 采用循环神经网络对标签之间的关系进行建模.

接下来将对上述算法的执行流程做一定阐述,对于词级模型,首先中文文书进行分词,算法 1) 2) 采用 TF-IDF 算法进行特征提取,算法 3) 4) 则将词嵌入至定长向量.

表 4　不同算法性能对比

Table 4　Performance comparison of different algorithms

算法	$MacroF_1\uparrow$	$MicroF_1\uparrow$	$RankL\downarrow$
BR	0.634	0.768	0.083
ML-KNN	0.645	0.771	0.095
CNN	0.691	0.819	0.065
CNN-RNN	0.695	0.825	0.061
本文的算法	0.738	0.846	0.048

表 4 展示了在全数据上,不同算法之间的性能比较. 图 3 展示了在不同比例数据集上的分类性能. 实验结果显示,随着数据规模的增大,深度学习算法能获得更好的表现. 相比于另两种深度模型 CNN 和 CNN-RNN,本文提出的算法的整体性能都较优. 这是由于研究在文本特征提取和标签决断上都做了考虑. 迁移的 BERT 模型能提供数据集之外的语义知识并且具有更多的参数量,由图 3 可见,模型表现受到数据集尺寸的波动较小. 配对排序损失能铺捉到标签之间的排序关系,使相关度较高的标签能获得更大的得分,同时,自适应的标签阈值学习能帮助算法得到更精准的预测结果.

6　总　结

多标签文本学习能帮助用户对文档进行有效管理,加强多媒体系统的可用性. 传统的,基于机器学习的算法受限于特征提取和模型容量,存在严重性能瓶颈. 本文提出的算法利

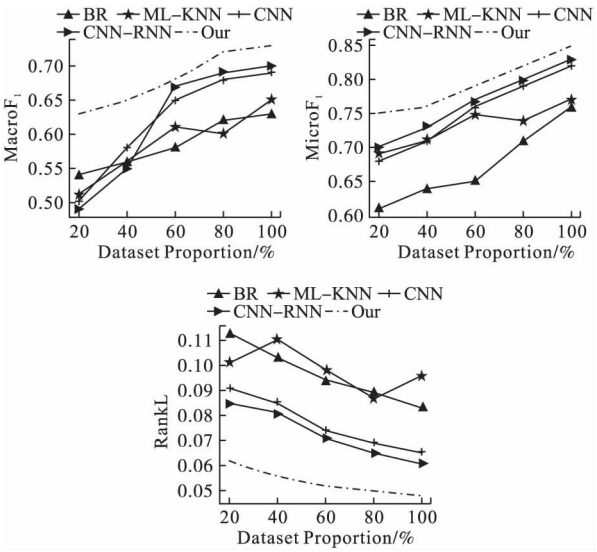

图 3　不同比例数据集上的对比

Fig. 3　Comparison with different dataset proportion

用中文 BERT 预训练语言模型对文书进行特征提取,模型架构上更精炼且高. 算法选用配对排序损失作为目标函数,以铺捉到标签之间的关系. 此外,为了更精准地得到结果,引入辅助的阈值预测模型,对标签预测进行建模. 实验在法条预测和罪名推荐两项任务上验证了算法的有效性. 作为自然语言处理的一项子任务,BERT 对多标签文本分类也是适用的,将阈值预测看作一项学习任务,相比 *Top-k* 和全局阈值,在测试集上表现更优异. 未来我们将在更多的多标签数据集上对算法进行验证,并将对标签之间的相关性做进一步探讨.

References:

[1] Devlin Jacob, Ming-Wei Chang, Kenton Lee, et al. Bert: pre-training of deep bidirectional transformers for language understanding [C] //Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 4171-4186.

[2] Zhang Wen-jie, Yan Jun-chi, Wang Xiang-feng, et al. Deep extreme multi-label learning [C] //Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ACM, 2018: 100-107.

[3] Pengcheng Yang, Xu Sun, Wei Li, et al. SGM: sequence generation model for multi-label classification [C] //Proceedings of the 27th International Conference on Computational Linguistics (COLING), 2018: 3915-3926.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need [C] //Advances in Neural Information Processing Systems, 2017: 5998-6008.

[5] Liu Jing-zhou, Wei-Cheng Chang, Yuexin Wu, et al. Deep learning for extreme multi-label text classification [C] //International Acm Sigir Conference on Research & Development in Information Retrieval, ACM, 2017: 115-124.

[6] Zhang Honglei, Serkan Kiranyaz, Moncef Gabbouj. A k-nearest neighbor multilabel ranking algorithm with application to content-based image retrieval [C] //2017 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2017: 2587-2591.

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

[8] Kurata, Gakuto, Bing Xiang, et al. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence [C] //Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016: 521-536.

[9] Gong Yunchao, Yangqing Jia, Thomas Leung, et al. Deep convolutional ranking for multilabel image annotation [C] //2nd International Conference on Learning Representations, 2014.

[10] NamJinseok, Jungi Kim, Eneldo Loza Mencía, et al. Large-scale multi-label text classification—revisiting neural networks [C] //Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2013: 437-452.

[11] Zhang Min-ling, Zhi-Hua Zhou. A review on multi-label learning algorithms [J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(8): 1819-1837.

[12] Tsoumakas, Grigorios, Ioannis Katakis, et al. Random k-labelsets for multilabel classification [J]. IEEE Transactions on Knowledge & Data Engineering, 2011, 23(7): 1079-1089.

[13] Zhang Min-ling, Zhi-Hua Zhou. ML-KNN: a lazy learning approach to multi-label learning [J]. Pattern Recognition, 2007, 40(7): 2038-2048.

[14] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo LozaMencía, et al. Multilabel classification via calibrated label ranking [J]. Machine Learning, 2008, 73(2): 133-153.

[15] Kim, Yoon. Convolutional neural networks for sentence classification [C] //Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1746-1751.

[16] Zhang Min-ling, Zhou Zhi-hua. Multilabel neural networks with applications to functional genomics and text categorization [J]. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(10): 1338-1351.

[17] Mikolov Tomas, Ilya Sutskever, Kai Chen, et al. Distributed representations of words and phrases and their compositionality [C] //Advances in Neural Information Processing Systems, 2013, 26: 3111-3119.

[18] Yuncheng Li, Yale Song, Jiebo Luo. Improving pairwise ranking for multi-label image classification [C] //Conference on Computer Vision and Pattern Recognition, 2017: 1837-1845.

[19] Han Fei, Chai Yu-mei, Wang Li-ming, et al. Text classification method combining random walk and rough decision [J]. Journal of Chinese Computer Systems, 2019, 40(6): 1165-1173.

[20] Yang Shuai-hua, Zhang Qing-hua. Research on k-nearest neighbor text classification algorithm of approximation set of rough set [J]. Journal of Chinese Computer Systems, 2017, 38(10): 2192-2196.

附中文参考文献:

[19] 韩 飞, 柴玉梅, 王黎明, 等. 一种结合随机游走和粗糙决策的文本分类方法 [J]. 小型微型计算机系统, 2019, 40(6): 1165-1173.

[20] 杨帅华, 张清华. 粗糙集近似集的 KNN 文本分类算法研究 [J]. 小型微型计算机系统, 2017, 38(10): 2192-2196.