计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(6):167−174 [doi: 10.15888/j.cnki.csa.008488]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合 BERT 和图注意力网络的多标签文本分类①

郝　超, 裘杭萍, 孙　毅

(陆军工程大学 指挥控制工程学院, 南京 210007)
通信作者: 裘杭萍, E-mail: qiuhp_zy@163.com

摘　要: 多标签文本分类问题是多标签分类的重要分支之一, 现有的方法往往忽视了标签之间的关系, 难以有效利用标签之间存在着的相关性, 从而影响分类效果. 基于此, 本文提出一种融合 BERT 和图注意力网络的模型 HBGA (hybrid BERT and graph attention): 首先, 利用 BERT 获得输入文本的上下文向量表示, 然后用 Bi-LSTM 和胶囊网络分别提取文本全局特征和局部特征, 通过特征融合方法构建文本特征向量, 同时, 通过图来建模标签之间的相关性, 用图中的节点表示标签的词嵌入, 通过图注意力网络将这些标签向量映射到一组相互依赖的分类器中, 最后, 将分类器应用到特征提取模块获得的文本特征进行端到端的训练, 综合分类器和特征信息得到最终的预测结果. 在 Reuters-21578 和 AAPD 两个数据集上面进行了对比实验, 实验结果表明, 本文模型在多标签文本分类任务上得到了有效的提升.

关键词: 多标签文本分类; 图注意力网络; BERT; 深度学习

引用格式: 郝超,裘杭萍,孙毅.融合 BERT 和图注意力网络的多标签文本分类.计算机系统应用,2022,31(6):167–174. http://www.c-s-a.org.cn/1003-3254/8488.html

Incorporating BERT and Graph Attention Network for Multi-label Text Classification

HAO Chao, QIU Hang-Ping, SUN Yi

(Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China)

Abstract: The multi-label text classification is one of the important branches of multi-label classification. Existing methods often ignore the relationship between labels, and thus the correlation between labels can hardly be put into effective use, which affects the effects of classification. On this basis, this study proposes a hybrid BERT and graph attention (HBGA) model that fuses BERT and the graph attention network. First, BERT is employed to obtain the context vector representation of the input text, and Bi-LSTM and the capsule network are used to extract the global and local features of the text, respectively. Then, through feature fusion, text feature vectors are constructed. Meanwhile, the correlation between labels is modeled through graphs, and the nodes in graphs are used to represent the word embedding of the labels, and these label vectors are mapped to a set of interdependent classifiers through the graph attention network. Finally, the classifiers are applied to the text features obtained by the feature extraction module for end-to-end training. The classifier and feature information are integrated to obtain the final prediction results. Comparative experiments are performed on datasets Reuters-21578 and AAPD, and the experimental results indicate that the model in this study has been effectively improved on tasks of multi-label text classification.

Key words: multi-label text classification; graph attention network; BERT; deep learning

① 收稿时间: 2021-08-13; 修改时间: 2021-09-13; 采用时间: 2021-09-22; csa 在线出版时间: 2022-05-26

在全球信息化大潮的推动下, 大数据得到了快速的发展, 人们处在一个海量数据的世界, 每时每刻都有新的数据信息产生, 这些数据不仅数量大并且还具有多样性, 这也使得人们用传统手段统计此类数据的时候变得困难[1]. 如何高效地处理这些数据是一个很有研究意义的问题, 这也推动着自动分类技术的发展. 传统的文本分类问题中每个样本只对应一个标签类别, 属于单标签文本分类. 但在现实生活中, 样本信息往往不够理想, 一个样本可能拥有更加复杂的语义和内容[2]. Schapire 等人[3] 提出了多标签学习, 与单标签文本分类不同, 多标签学习指的是从标签集中为每个样本分配最相关的标签子集的过程, 从而能够更加准确地、有效地表示单标签文本分类中不能表达复杂语义和内容. 比如题为“打造特色体育教学, 推进阳光体育运动”的新闻可能被同时认为与“体育”和“教育”两者相关, 一条微博可能同时与“新冠”“疫苗”和“医疗”有关等等.

1 相关工作

目前, 有关多标签文本分类已经提出很多方法, 这些方法主要可以分为 3 大类: 问题转换方法、算法自适应方法和基于深度学习方法.

问题转换方法是最经典的方法, 通过将多标签分类问题转化为多个单标签分类问题来解决, 代表性的方法包括二元相关 (binary relevance, BR)[4]、标签幂集分解 (label powerset, LP)[5] 和分类器链 (classifier chain, CC)[6]. BR 方法将多标签分类问题分解为多个二分类问题来进行处理; LP 方法通过将标签组合看成分类类别, 将多标签分类问题转化为多分类问题来处理; CC 方法将多标签分类任务转化为二进制分类问题链, 后续的二进制分类器链基于前面的进行预测.

算法自适应方法通过扩展相应的机器学习方法来直接处理多标签分类问题, 代表性的方法包括 ML-DT (multi-label decision tree)、排名支持向量机 (ranking support vector machine, Rank-SVM) 和多标签 K 最近邻 (multi-label K-nearest-neighborhood, ML-KNN). ML-DT 方法通过构造决策树来执行分类操作; Rank-SVM 方法通过支持向量机 (support vector machine, SVM) 来处理多标签分类问题; ML-KNN 方法通过改进 KNN 方法以实现通过 K 近邻来处理多标签数据.

随着深度学习的发展许多基于深度学习的多标签文本分类方法被提出, 代表性的方法包括 TextCNN[7]、XML-CNN[8]、CNN-RNN[9]、SGM[10] 和 MAGNET[11]. TextCNN 首次将 CNN 应用于文本分类; XML-CNN 方法是对 TextCNN 方法的改进, 采用了动态池化和二元交叉熵损失函数; CNN-RNN 方法通过将 CNN 和 RNN 进行融合来实现多标签分类; SGM 方法采用 Seq2Seq 结构, 首次将序列生成的思想应用到多标签文本分类中; MAGNET 方法利用 Bi-LSTM 提取文本的特征, 用图神经网络构建各标签之间的内在联系.

现有的方法没能充分考虑标签之间的相关性, 从而影响了分类效果. 针对此问题, 本文提出了一种基于 BERT 和图注意力网络 (graph attention network, GAT) 的模型, 主要利用 BERT 模型获得文本的上下文表示, 通过 Bi-LSTM 和胶囊网络分别提取全局特征和局部特征, 利用 GAT 捕获标签之间的相关性, 从而来提升分类的性能.

2 模型构建

多标签文本分类 (multi-label text classification, MLTC) 的主要任务是通过若干类别标签对文本样本进行标注分类, 可形式化描述: d 维的实例空间 $X=\mathbb{R}^d$, q 个标签组成的标签空间 $Y=\{y_1,y_2,y_3,\cdots,y_q\}$, 训练集为 $D=\{(x_i,Y_i)|1\leqslant i\leqslant m\}$, 模型通过从实例空间到标签空间学习一个映射: $h:X\to 2^Y$ 多标签文本分类任务. 其中, 在每个实例 (x_i,Y_i) 中, $x_i\in X$ 是 d 维特征向量, $Y_i\subseteq Y$ 是实例 x_i 的标签集合, 测试样本通过映射 h 便可得到相应的标签集合[12].

本文提出的模型主要包括 BERT 模块、特征提取与融合模块、GAT 分类器模块 3 个部分, 具体的框架如图 1 所示.

2.1 BERT 模块

文本信息对于人而言是可以直观理解的, 但是对于计算机而言无法直接处理, 因此需要将文本转化为计算机能够处理的数据. 传统的文本表示包括 one-hot 和矩阵分解, 但是在表示的时候会产生维度灾难、花费代价高等缺点, 随着神经网络的发展, 词嵌入 (word embedding) 作为一种新的词表示方式出现, 使用一个连续、低维、稠密的向量来表示单词, 简称为词向量.Word2Vec[13] 和 Glove[14] 是一种静态的词向量表示方式, 对于任意一个词, 其向量是恒定的, 不随其上下文的变化而变化. 比如“apple”一词在“APPLE Inc”和

“apple tree”中有同样的词向量, 但是这两处的意思是明显不一样的, 一个代表的是苹果科技公司, 一个代表苹果树, 静态词向量无法解决一词多义的问题.

为了更好的文本, 本文采用了预训练模型 BERT[15] 来计算每个单词的上下文表示, 依据不同的上下文对同一个单词有不同的表示, BERT 模型由多层 Transformer 构成, 接受 512 个词的序列输入, 并输出该序列的表示, 流程如图 2 所示. 对于由k个词组成的文档作为输入$W=[w_1,w_2,\cdots,w_k]$, 经过 BERT 模型得到相对应的词向量$E=[e_1,e_2,\cdots,e_k]$.

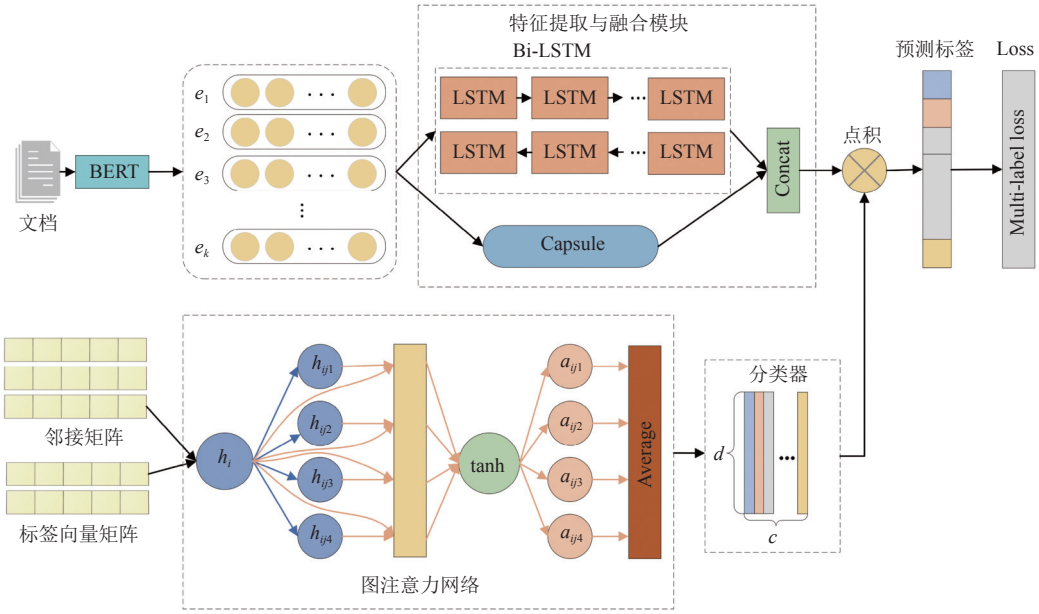

图 1 模型框架

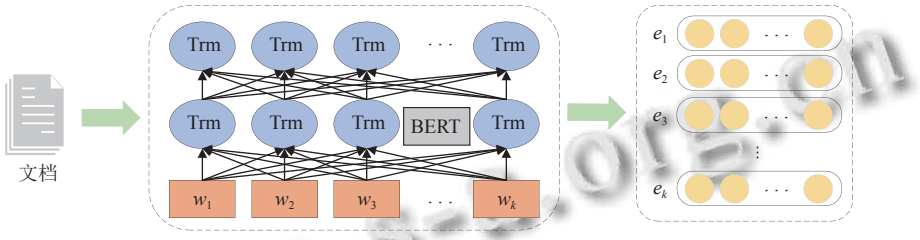

图 2 BERT 模块

2.2 GAT 分类器

在图卷积网络 (graph convolutional network, GCN) 中, 一个节点的相邻节点具有相同的权重, 然而在图结构中相邻节点的重要性存在一定差异. 在 GAT 中引入“注意力机制”[16] 对此缺点进行改进, 通过计算当前节点和相邻节点的“注意力系数”, 在聚合相邻节点时进行加权, 使得当前节点更加关注重要的节点. 因此, 本文采用了 GAT[17], 利用图注意力训练得到的结果作为该模型的分类器, 以便更好地挖掘标签之间的相关性. 结构如图 3 所示.

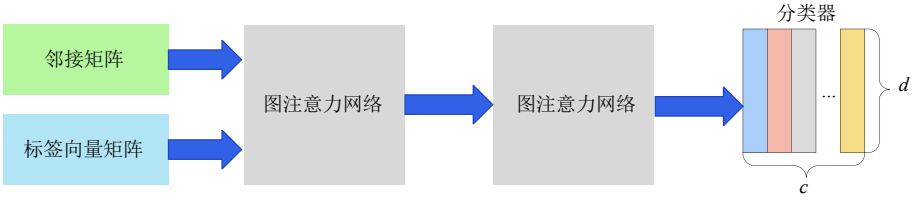

图 3 GAT 分类器

在此模块中, 将标签向量矩阵和邻接矩阵作为 GAT 输入, 经过两层的 GAT 得到最终的分类器. 标签向量采用 Stanford 官方预训练好的 Glove 词向量, 其中包括有 50 维、100 维和 300 维 3 种, 为了使标签包含更多的语义信息, 本文采用 300 维的 Glove 词向量作为 GAT 的输入. 通过数据驱动[18] 的方式建立邻接矩阵, 首先获得训练数据集中的标签共现矩阵$M \in \mathbb{R}^{c\times c}$, 频率向量$F \in \mathbb{R}^{c}$, F_i表示的就是标签i在训练集中出现的频率, 其中, c代表的是标签数量. 通过以下计算能够获得初始邻接矩阵A:

$$A = \frac{M}{F} \tag{1}$$

用如下公式更新l层的每个节点i的向量表示, 其中W是一个训练参数:

$$h^{(l+1)} = \sigma(Ah^lW^l) \tag{2}$$

在 GAT 中, “注意力系数”是一个非常重要的指标. “注意力系数”$\alpha_{ij}^{(l)}$代表网络在更新第l层的i节点时, j节点对其的重要程度. 其基本的表达式如下:

$$\begin{aligned}\alpha_{ij}^{(l)} &= f(h_i^{(l)}W^{(l)}, h_j^{(l)}W^{(l)}) \\ &= LeakyReLU((h_i^{(l)}W^{(l)}) \,\|\, (h_j^{(l)}W^{(l)})^T)\end{aligned} \tag{3}$$

其中, $\|$代表连接操作.

在本文模型中, 采用了 Vaswani 等人[16] 提到的多头注意力, 通过不同的注意力来获得更多的标签间的关系. 此操作将被复制 K 次, 每一次的参数都是不相同的, 最终将 K 次结果求均值得到最终的输出, 其计算公式如下:

$$h_i^{(l+1)} = \tanh\left(\frac{1}{k}\sum_{k=1}^{K}\sum_{j\in N(i)}\alpha_{i,j,k}^{l}h_j^lW^l\right) \tag{4}$$

其中, $N(i)$表示i的相邻节点的个数. 采用了 GAT 层的级联操作, 在第一层, 输入的是标签向量矩阵$M \in \mathbb{R}^{c\times d}$, 可以用如下公式表示:

$$h_i^1 = \tanh\left(\frac{1}{k}\sum_{k=1}^{K}\sum_{j\in N(i)}\alpha_{i,j,k}^{0}MW^{(0)}\right) \tag{5}$$

与 RNN 类似, GAT 将上一层的输出传入到下一层的输入, 但是 GAT 的权重之间不共享, 最终得到输出$H_{\text{gat}} = (h_1, h_2, \cdots, h_c) \in \mathbb{R}^{c\times d}$, c代表标签的数量, d代表标签的维度.

2.3 特征提取与融合模块

将 BERT 模块得到词向量分别作为 Bi-LSTM 和胶囊网络的输入, 之后进行特征提取. 在特征提取时采用 Bi-LSTM 来提取全局特征, 并通过胶囊网络来兼顾局部特征, 最后通过特征融合的方式得到最终的特征向量. 这样能够充分利用上下文信息, 减少特征的丢失, 从而带来更好的分类效果.

(1) 胶囊网络

通过对卷积神经网络进行改进形成了胶囊网络, 在传统的卷积神经网络中, 池化步骤往往采用的是最大池化或者平均池化, 此过程中会造成特征信息大量丢失. 针对这一问题, Hinton 提出的胶囊网络[19] 用神经元向量代替卷积神经网络中的单个神经元节点, 能够确保保存更多的特征信息, 提取到局部特征.

动态路由是胶囊网络的核心机制, 通过动态路由来训练神经网络, 能够获取文本序列中的单词位置信息并捕获文本的局部空间特征, 动态路由的过程如图 4 所示.

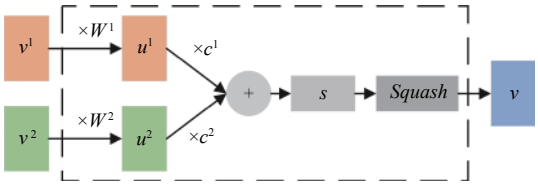

图 4 胶囊网络

在胶囊网络中, 底层胶囊u_i将输入向量传递到高层胶囊$\hat{u}_{j|i}$的过程称为路由, 高层胶囊和底层胶囊的权重通过动态路由获得, 过程如下:

$$\hat{u}_{j|i} = W_{ij}u_i \tag{6}$$

$$c_{ij} = Softmax(b_{ij}) = \frac{\exp(b_{ij})}{\sum_j \exp(b_{ij})} \tag{7}$$

其中, W_{ij}为权重矩阵; c_{ij}为耦合系数.

c_{ij}用来预测上一层胶囊和下一层胶囊的相似性, 其通过动态路由的过程来决定, 并且输入层和输出层之间所有c_{ij}值和为 1; b_{ij}的初始值设置为 0, 通过迭代更新.

$$s_j = \sum_i c_{ij}\cdot\hat{u}_{j|i} \tag{8}$$

传统的神经网络中, 多数情况下会使用 Sigmoid、tanh 和 ReLU 等激活函数, 但在胶囊网络中创建了一个新的激活函数 *Squash*, 只会改变向量的长度, 不会改

变向量的方向. 其中, s_j通过耦合系数c_{ij}和$\hat{u}_{j|i}$加权求和来得到, 作为 *Squash* 函数的输入.

$$Squash(x)=\frac{\|x\|^2}{1+\|x\|^2}\frac{x}{\|x\|} \tag{9}$$

$$v_j=Squash(s_j) \tag{10}$$

$$b_{ij}\leftarrow b_{ij}+\hat{u}_{j|i}\cdot v_j \tag{11}$$

通过胶囊网络动态路由的迭代, 可以获得局部特征$H_c=(v_1,v_2,v_3,\cdots,v_k)$.

(2) Bi-LSTM

长短时记忆网络 (long short-term memory, LSTM)[20] 能够有效缓解梯度消失问题, 但 RNN 和 LSTM 都只能依据前一时刻的信息来预测下一时刻的输出. 在有些问题中, 当前时刻的输出除了与之前的状态有关外, 还可能和未来的状态有一定的联系. 比如在对缺失单词进行预测时, 往往需要将其上下文同时考虑才能获得最准确的结果.

双向长短时记忆网络 (bi-directional long short-term memory, Bi-LSTM)[21] 有效地改善了这一问题. 一个前向的 LSTM 和一个后向的 LSTM 组合成 Bi-LSTM. 通过前向和后向的特征提取, 能够更好地建立上下文之间的关系, 从而捕获全局文本特征. Bi-LSTM 结构如图 5 所示, 计算公式如下:

$$\overrightarrow{h}_i=\overrightarrow{LSTM}(\overrightarrow{h}_{i-1},x_i) \tag{12}$$

$$\overleftarrow{h}_i=\overleftarrow{LSTM}(\overleftarrow{h}_{i+1},x_i) \tag{13}$$

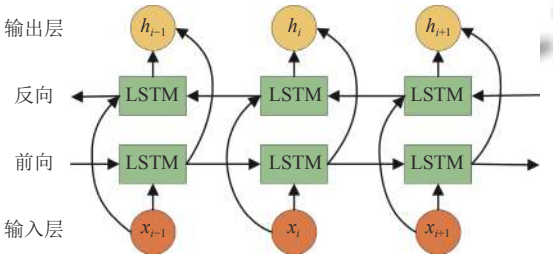

图 5 Bi-LSTM 结构

通过拼接前向和后向 LSTM 的输出$\overrightarrow{h}_i$和$\overleftarrow{h}_i$可以获得第i个单词的最终隐藏层输出$h_i=[\overrightarrow{h}_i,\overleftarrow{h}_i]$, 通过对于前文信息和后文信息进行处理, 挖掘输入序列的上下文信息, 最终得到向量$H_L=(h_1,h_2,h_3,\cdots,h_k)$.

(3) 特征融合

传统的模式识别通常运用的是人工设计的特征, 经过特征提取算法得到特征数据; 神经网络相比与传统模式识别, 具有自动提取特征的特点和更好的特征提取效果, 特征的优劣影响分类结果的好坏. 因此, 需要提取较优的特征. 胶囊网络作为对卷积神经网络的改进, 在提取局部特征的时候有着不错的效果; Bi-LSTM 通过前向和后向的传播, 能够较好关联上下文信息, 提取全局特征.

在本文模型中, 充分的发挥两者的优势, 分别采用胶囊网络和 Bi-LSTM 来提取文本的局部和全局的特征, 通过特征融合的方式将两者特征提取结果进行信息的融合连接. 融合连接有两种常用方式: 第一种是相加融合, 通过直接将对应维度的数据进行相加实现, 这种方式不会改变特征向量的维度, 能够避免维数灾难; 另一种是拼接融合, 通过将维度进行拼接来实现, 这种方式会是将两种特征向量拼接后, 维度会变大[22]. 由于拼接融合会导致维度增大, 可能会造成维度灾难, 因此, 本文选择相加融合的方式.

胶囊网络提取的特征可以用$H_c=(v_1,v_2,v_3,\cdots,v_k)$来表示, Bi-LSTM 提取的特征可以用$H_L=(h_1,h_2,h_3,\cdots,h_k)$来表示, 通过相加融合的方式, 可以得到新的特征$H$, 从而提升模型的效果.

通过分类器训练获得的每个标签向量和胶囊网络以及 Bi-LSTM 获得的融合特征向量相乘就可以得到标签最终的得分, 得到最终的结果. 计算公式如下:

$$\hat{y}=H\odot H_{\text{gat}} \tag{14}$$

2.4 损失函数

在实验中, 损失函数选择二元交叉熵 (binary cross entropy loss), 它广泛应用于神经网络分类训练任务中. 假设文本的真实值是$y\in\mathbb{R}^c$, $y^i=\{0,1\}$表示标签i是否属于该文本, $\hat{y}^c$表示的是模型的预测值. 具体的计算公式如下:

$$L=\sum_{c=1}^{C}(y^c\ln(\sigma(\hat{y}^c))+(1-y^c)\ln(1-\sigma(\hat{y}^c))) \tag{15}$$

其中, $\sigma(\cdot)$代表的是 Sigmoid 函数.

3 实验与结果分析

3.1 数据集介绍

本文采用了多标签文本分类领域常用的数据集. 包括 Reuters-21578 和 AAPD, 表 1 为数据集详细信息.

Reuters-21578: 该数据集是由路透社新闻组成的,

收集了 10 788 条来自路透社的新闻, 包括 7 769 条训练集和 3 019 条测试集组成, 一共包含 90 个类别.

AAPD[9]: 该数据集是由 Yang 等人提供. 是从网络上收集了 55 840 篇论文的摘要和相应学科类别, 一篇学术论文属于一个或者多个学科, 总共由 54 个学科组成.

表 1 数据集简介

数据集	样本总数	标签数	训练集	测试集
Reuters-21578	10 788	90	7 769	3 019
AAPD	55 840	54	44 672	11 168

3.2 实验参数设置

本实验利用了 GAT 来捕获标签之间的关系, 在进行实验时, 主要采用了两层带有多头注意力的 GAT 层. 在对句子和标签进行表示时候, 均采用了 BERT 向量获得其表示. 对于 Reuters-21578 和 AAPD 数据集, 模型的批处理大小 Batch Size 均设置为 250, 训练过程中使用了 Adam 优化器来使目标函数最小化, 学习率大小 Learning Rate 设置为 0.001, 并且在模型中添加了 Dropout 层来防止过拟合, Dropout 的值取 0.5, 多头注意力机制头的个数 K=8. 表 2 为实验参数的汇总.

表 2 网络参数说明表

参数名	参数值
批量大小	250
学习率	0.001
向量维度	768
隐藏层维度	250

3.3 实验评价指标

在本文的实验中, 使用 *Micro-precision*、*Micro-recall*、*Micro-F*1[23] 和汉明损失[3] 作为评价指标, 其中, 将 *Micro-F*1 作为主要的评价指标, 各个指标的具体计算公式如下:

$$Micro\text{-}precision = \frac{\sum_{j=1}^{L} TP_j}{\sum_{j=1}^{L} (TP_j + FP_j)} \tag{16}$$

$$Micro\text{-}recall = \frac{\sum_{j=1}^{L} TP_j}{\sum_{j=1}^{L} (TP_j + FN_j)} \tag{17}$$

$$Micro\text{-}F1 = \frac{\sum_{j=1}^{L} 2TP_j}{\sum_{j=1}^{L} (2TP_j + FP_j + FN_j)} \tag{18}$$

其中, L 代表类别标签数量, TP 代表原来是正样本被预测为正的数量, FP 代表原来是正样本被预测为负的数量, FN 代表原来是负样本被预测为正的数量.

汉明损失指的是被错分的标签的比例大小, 也就是两个标签集合的差别占比. 其计算公式如下:

$$HL = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{XOR(x_i, y_i)}{|L|} \tag{19}$$

其中, $|S|$是样本的数量, $|L|$是标签的总数, x_i表示标签, y_i表示真实标签, XOR 是异或运算.

3.4 实验结果与分析

3.4.1 实验对比

为了验证本文提出模型的有效性, 选择与现有的多标签文本分类方法: BR[4]、CC[6]、ML-KNN[24]、CNN[7]、CNN-RNN[9]、S2S+Attn[25]、MAGNET[11] 进行对比实验.

本文提出的方法在 Reuters-21578 和 AAPD 数据集的结果如表 3 和表 4. 在正确率 (P)、召回率 (R)、$F1$ 值和汉明损失 (HL) 4 个常用的评价指标上与其他模型进行了对比, P、R 和 $F1$ 中的“+”代表该值越高, 模型的效果越好, HL 这一列中的“–”代表该值越小, 模型的效果越好. 其中实验的最佳结果由加粗黑体表示.

表 3 Reuters-21578 数据集上结果对比

模型方法	P (+)	R (+)	$F1$ (+)	HL (–)
BR	0.930	0.816	0.870	0.003 2
CC	0.928	0.812	0.867	0.003 1
ML-KNN	0.803	0.473	0.595	0.008 8
CNN-RNN	0.902	0.813	0.855	0.003 7
S2S+Attn	0.916	0.818	0.864	0.003 4
MAGNET	—	—	0.890	0.002 9
Ours	**0.931**	**0.858**	**0.893**	**0.002 5**

表 4 AAPD 数据集上结果对比

模型方法	P (+)	R (+)	$F1$ (+)	HL (–)
BR	0.644	0.648	0.646	0.031 6
CC	0.657	0.651	0.654	0.030 6
ML-KNN	0.672	0.614	0.642	0.030 1
CNN	**0.849**	0.545	0.664	0.028 7
CNN-RNN	0.718	0.618	0.669	0.028 2
S2S+Attn	0.720	0.639	0.677	0.026 1
MAGNET	—	—	0.696	0.025 2
Ours	0.749	**0.676**	**0.711**	**0.024 5**

从表 3 和表 4 的实验结果可以看出, 本文提出的模型在 Reuters-21578和 AAPD 数据集上大部分评价指标上都展示了最好的结果. 在 Reuters-21578 数据集上, 与 CNN-RNN 相比在 *F*1 值上面提升了接近 4%, 汉明损失的值也取得了最优的结果; 在 AAPD 数据集上, 本文模型在召回率、*F*1 值和 HL 值上相比于其他模型均达到了最佳效果, 其中 *F*1 值比最优模型 MAGNET 提升了约 1.5%. 在准确率指标上, 传统的 CNN 表现最佳, 本文模型次之. 主要原因在于 CNN 是基于字符级别的模型, 利用网络特点细粒度地抓取标签与字符文本之间的关联, 从而提高模型的准确率, 另外, 在实验训练方面, CNN 在分类任务上超参数调整较小, 也是其在准确率上取得最佳表现的原因之一. 综合 4 类评价指标的实验结果来看, 本文提出的模型比其他模型更具适用性, 在有效提升 *F*1 值、召回率和减少汉明损失的同时, 兼顾了多标签文本分类的准确率.

从实验结果看, 深度学习方法普遍要比传统机器学习方法 (包括 BR、CC、LP 等方法) 表现更好. 这主要是由于传统机器学习方法处理此类问题的时候是利用人来提取特征, 往往会带来一些误差, 并且在一些复杂情况下, 有更多的局限性. 而深度学习方法最大的进步就是能够自动提取特征, 从而比传统机器学习方法有更好的效果, 在特征提取上, 深度学习领域也涌现出了很多方法, 本文提出的方法采用了 Bi-LSTM 和胶囊网络的方法, 比只采用了 Bi-LSTM 的模型 MAGNET 有着更好的效果, 证明了胶囊网络的有效性.

在处理多标签文本分类问题的时候, 标签之间的相关性是非常重要的信息之一. 传统的机器学习模型在处理多标签文本分类问题上没有考虑标签之间的相关性, 本文提出的方法利用 GAT 来捕获标签之间的相关性并建模, 从而来生成分类器, 提升了在多标签文本分类任务上的效果.

综上可知, 本文提出的方法与传统机器学习方法和现有的深度学习方法相比, 取得了具有竞争力的结果.

3.4.2 不同词向量比较

为了验证 BERT 在词向量上的表现, 采用了一组对比实验来说明. 采用了目前比较常用的 3 种词向量包括 Word2Vec 向量、Glove 向量和 BERT 向量, 并且在对比实验中加入随机向量 (random). 在 Reuters-21578 数据集上进行比较, 结果如图 6 所示.

从图 6 可以看出, Word2Vec 向量和 Glove 向量的结果接近, 随机向量的结果是最差的, BERT 向量的结果是最好的, 因此, 用 BERT 向量能够提升本文方法的准确率.

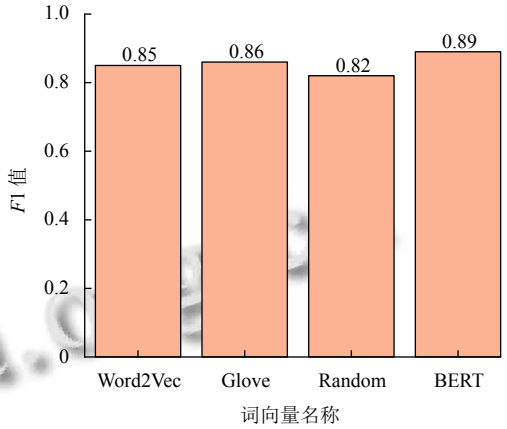

图 6 Reuters-21578 数据集词向量比较

4 总结

本文提出了一种基于 BERT 和 GAT 的模型 HBGA 来解决多标签文本分类问题, 该模型是一个端到端的结构. 首先, 利用 BERT 模型获取文本的上下文向量, 通过 GAT 来捕获标签之间的注意力依赖结构, 使用特征矩阵和邻接矩阵来探索标签之间的关系, 进行训练后形成一个分类器, 利用 Bi-LSTM 和胶囊网络分别提取文本的全局特征和局部特征, 进行特征融合获得文本的特征向量, 最后将分类器和特征向量进行整合得到最终的结果. 实验结果表明, 提出的模型在 *F*1 值上均优于对比模型, 有效地提升了多标签文本分类的性能. 目前模型仅仅在标签集小的数据集下取得不错的效果, 在接下来的工作中, 将探究如何在大规模标签集下的提升性能.

参考文献

1 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. 软件学报, 2020, 31(4): 1079–1089. [doi: 10.13328/j.cnki.jos.005923]

2 郝超, 裘杭萍, 孙毅, 等. 多标签文本分类研究进展. 计算机工程与应用, 2021, 57(10): 48–56. [doi: 10.3778/j.issn.1002-8331.2101-0096]

3 Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Machine Learning, 1999, 37(3): 297–336. [doi: 10.1023/A:1007614523901]

4 Boutell MR, Luo JB, Shen XP, *et al*. Learning multi-label scene classification. Pattern Recognition, 2004, 37(9): 1757–1771. [doi: 10.1016/j.patcog.2004.03.009]

5 Tsoumakas G, Katakis I. Multi-label classification: An overview. International Journal of Data Warehousing and Mining (IJDWM), 2007, 3(3): 1–13. [doi: 10.4018/jdwm.2007070101]

6 Read J, Pfahringer B, Holmes G, *et al*. Classifier chains for multi-label classification. Machine Learning, 2011, 85(3): 333–359. [doi: 10.1007/s10994-011-5256-5]

7 Kim Y. Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha: EMNLP, 2014. 1746–1751.

8 Liu JZ, Chang WC, Wu YX, *et al*. Deep learning for extreme multi-label text classification. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. Shinjuku: ACM, 2017. 115–124.

9 Chen GB, Ye DH, Xing ZC, *et al*. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. 2017 International Joint Conference on Neural Networks (IJCNN). Anchorage: IEEE, 2017. 2377–2383.

10 Yang PC, Sun X, Li W, *et al*. SGM: Sequence generation model for multi-label classification. arXiv: 1806.04822, 2018.

11 Pal A, Selvakumar M, Sankarasubbu M. MAGNET: Multi-label text classification using attention-based graph neural network. Proceedings of the 12th International Conference on Agents and Artificial Intelligence, Volume 2: ICAART. Valletta: ICAART, 2020. 494–505.

12 Zhang ML, Zhou ZH. A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819–1837. [doi: 10.1109/TKDE.2013.39]

13 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv: 1301.3781, 2013.

14 Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha: ACL, 2014. 1532–1543.

15 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv: 1810.04805, 2018.

16 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

17 Veličković P, Cucurull G, Casanova A, *et al*. Graph attention networks. arXiv: 1710.10903, 2017.

18 Chen ZM, Wei XS, Wang P, *et al*. Multi-label image recognition with graph convolutional networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 5172–5181.

19 Sabour S, Frosst N, Hinton GE. Dynamic routing between capsules. arXiv: 1710.09829, 2017.

20 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

21 Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks, 2005, 18(5-6): 602–610. [doi: 10.1016/j.neunet.2005.06.042]

22 刘心惠, 陈文实, 周爱, 等. 基于联合模型的多标签文本分类研究. 计算机工程与应用, 2020, 56(14): 111–117. [doi: 10.3778/j.issn.1002-8331.1904-0273]

23 Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press, 2008.

24 Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition, 2007, 40(7): 2038–2048. [doi: 10.1016/j.patcog.2006.12.019]

25 Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2014. 3104–3112.