计算机科学
COMPUTER SCIENCE

基于并行卷积网络信息融合的层级多标签文本分类算法

易流, 耿新宇, 白静

引用本文

易流, 耿新宇, 白静. 基于并行卷积网络信息融合的层级多标签文本分类算法[J]. 计算机科学, 2023, 50(9): 278-286.

YI Liu, GENG Xinyu, BAI Jing. Hierarchical Multi-label Text Classification Algorithm Based on Parallel Convolutional Network Information Fusion [J]. Computer Science, 2023, 50(9): 278-286.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221200133

基于并行卷积网络信息融合的层级多标签文本分类算法

易 流 耿新宇 白 静
西南石油大学计算机科学学院 成都 610000
(y1u6rk@163.com)

摘 要 自然语言处理是人工智能与机器学习领域的重要方向,它的目标是利用计算机技术来分析、理解和处理自然语言。自然语言处理的一个重点研究方向是从文本内容中获取信息,并且按照一定的标签体系或标准将文本内容进行自动分类标记。相比于单一标签文本分类而言,多标签文本分类具有一条数据属于多个标签的特点,使得更难从文本信息中获得多类别的数据特征。层级多标签文本分类又是其中的一个特别的类别,它将文本中的信息对应划分到不同的类别标签体系中,各个类别标签体系又具有互相依赖的层级关系。因此,如何利用其内部标签体系中的层级关系更准确地将文本分类到对应的标签中,也就成了解决问题的关键。为此,提出了一种基于并行卷积网络信息融合的层级多标签文本分类算法。首先,该算法利用 BERT 模型对文本信息进行词嵌入,接着利用自注意力机制增强文本信息的语义特征,然后利用不同卷积核对文本数据特征进行抽取。通过使用阈值控制树形结构建立上下位的节点间关系,更有效地利用了文本的多方位语义信息实现层级多标签文本分类任务。在公开数据集 Kanshan-Cup 和 CI 企业信息数据集上的结果表明,该算法在宏准确率、宏召回率与微 F1 值 3 种评价指标上均优于主流的 TextCNN,TextRNN,FastText 等对比模型,具有较好的层级多标签文本分类效果。

关键词:层级多标签文本分类;预训练模型;注意力机制;卷积神经网络;树形结构
中图法分类号 TP391

Hierarchical Multi-label Text Classification Algorithm Based on Parallel Convolutional Network Information Fusion

YI Liu,GENG Xinyu and BAI Jing
School of Computer Science,Southwest Petroleum University,Chengdu 610000,China

Abstract Natural language processing(NLP) is an important research direction in the field of artificial intelligence and machine learning,which aims to use computer technology to analyze,understand,and process natural language. One of the main research areas in NLP is to obtain information from textual content and automatically classify and label textual content based on a certain labeling system or standard. Compared to single-label text classification,multi-label text classification has the characteristic that a data element belongs to multiple labels,which makes it more difficult to obtain multiple categories of data features from textual information. Hierarchical classification of multi-label texts isa special category,whichdivides the information contained in the text into different category labeling systems,and each category labeling system has an interdependent hierarchical relationship. Therefore,the use of the hierarchical relationship in the internal labeling system to more accurately classify the text into corresponding labels becomes the key to solving the problem. To this end,this paper proposes a hierarchical classification algorithm for multi-label texts based on the fusion of parallel convolutional network information. First,the algorithm uses the BERT model for word integration in textual information,then it enhances the semantic features of textual information using a self-attention mechanism and extracts the features of textual data using different convolutional kernels. The multi-faceted semantic information of the text is more effectively used for the task of a hierarchical classification of multi-label texts by using a threshold-controlled tree structure to establish inter-node relationships between higher and lower bits. The results obtained on the Kanshan-Cup public dataset and the CI enterprise information dataset show that the algorithm outperforms TextCNN,TextRNN,FastTex and other comparative models in three evaluation measures,namely macro-precision,macro-recall,and micro F1 value,and has a better cascade multi-label text classification effect.

Keywords Hierarchical multi-label text classification,Pre-training model,Attention mechanism,Convolutional neural network,Tree structure

到稿日期:2022-12-23 返修日期:2023-04-07
基金项目:四川省科技计划项目(2022NSFSC0555)
This work was supported by the Sichuan Science and Technology Program(2022NSFSC0555).
通信作者:耿新宇(gengxy123@126.com)

1　引言

自然语言处理(Natural Language Processing)是人工智能与机器学习领域的重要方向。自然语言处理中的一个重点研究方向是从文本内容获取信息并且按照一定的标签体系或标准将文本内容进行自动分类标记。文本分类被广泛应用在情感分析[1]、垃圾短信分类[2]、问答系统[3]、信息检索[4]等实际应用任务中。相比于单一标签的文本分类任务[5]，多标签文本分类的实际任务应用场景更为广泛[6]。层级多标签的文本分类任务是基于多标签文本分类任务的一个重要分支[7]，需要在对数据进行分类的同时，利用文本特征信息特征的层级关系，更加准确地将对应文本划分到具有上下位层级关系的标签体系当中[8]。

现阶段，为了提升文本语义的表示性并在模型中利用层级标签预测[9]，研究者们使用了基于深度学习的方法、基于多种词嵌入组合的 CNN 模型方法和基于预训练模型的方法去完成现阶段的层级多标签文本分类任务。

1)基于深度学习与图卷积的方法

(1)基于深度学习的方法。Liu 等[10]使用多标签文本分类中经典的深度学习算法，例如 Text Convolutional Neural Networks(TextCNN)[11]，Text Recurrent Neural Networks(TextRNN)[12]和 FastText[13]，对文本进行特征抽取，并对全连接层进行标签集大小适应。以上的深度学习模型均可以在一定程度上解决极限多标签文本分类问题，也可以提升文本对应的语义表示。

(2)基于图卷积的方法。TextGCN 和 GAT 等模型可以对文本信息进行文本图的构建。图卷积模型利用图中节点之间的关系构建特征图上的边，建立文本图，并利用文本图中单词节点与边的关系，完成文本分类的任务。

2)基于多种词嵌入组合的模型方法

Gargiulo 等[14]提出多种词嵌入组合的 CNN 方法，并利用预测标签及其所有祖先标签的扩展层级标签方法，进一步解决层次多标签文本分类问题。

Zheng 等[15]提出了一种用于多标签文本分类的 BLSTM_MLPCNN 模型，该模型联合字符向量与词向量作为模型输入，采用 BLSTM 模型构建文档特征图，最后使用多层感知器神经网络 MLPCNN 进行特征提取。实验结果表明，相比 CNN，RNN 以及两者的组合模型，BLSTM_MLPCNN 具有更高的分类精度。

3)基于预训练模型的方法

Duan 等[16]提出了基于 BERT 的中文多标签文本分类模型，将 BERT 表示的特征向量直接输入到 softmax 层进行分类。Lan 等[17]提出的 AL-BERT 预训练模型，采用矩阵分解和跨层参数共享技术对 BERT 模型进行参数压缩，在维持 BERT 性能的同时，降低了其模型的空间复杂度，并提高了模型的训练速度，同时又对模型进行了扩展优化。

基于当前的层级多标签文本分类任务与文本分类任务中所出现的不同范围的数据，研究者们提出了以下 3 种主流的方法[18]。

1)全局法：利用分类器同时处理所有的类别。

2)展平法：将原始问题分解为一组扁平的多标签分类子问题，从而忽略了文本与层次结构之间的关联以及不同层次结构之间的依赖关系，这也将导致错误传播和无法将其文本正确分类。

3)局部方法：为层级标签的每一个节点都设置一个分类器，一般使用机器学习分类器模型，如 SVM 等。

以上 3 种方法在进行层级多标签的文本分类任务时，均存在各自的不足。这将导致模型不能很好地解决对层级标签的信息提取和对文本数据的特征抽取不充分的问题，也会使得在模型中无法利用上层任务指导辅助下层的文本分类预测任务。

现阶段，多标签文本分类任务还面临着长尾标签将导致模型的预测无法获得较高准确率的问题。为了提高对底层长尾标签的预测性能[19]，研究者们主要提出了以下 3 类方法：利用机器学习的方法、利用数据增强的方法，以及基于知识转移的方法。这 3 类方法旨在增加模型对尾部标签的优先级或扩充数据并增强数据标签，不仅可以充分利用长尾标签的数据信息，也可以提升对底层长尾标签的预测性能。

1)利用机器学习的方法

Cai 等[20]提出了一种基于支持向量机的层次分类方法来解决层级多标签文本分类任务，但这些传统的机器学习方法往往是基于词袋模型在文本中对其语义进行建模，在一定程度上限制了分类任务的预测性能。

2)利用数据增强的方法

对底层长尾标签生成更多的数据，分为扩充数据和增强标签两种方法。其中，扩充数据是增加模型数据的样本数据量。而增强标签则是利用标签之间的相关性来解决长尾问题。标签增强主要有以下两种思路：

(1)减少标签之间的竞争关系，增加尾部标签的优先级。

(2)利用图卷积的方式对标签之间的关系进行建模。该类方法通过增加尾部标签之间的相关性，促使更好地对标签信息进行增强。

3)基于知识转移的方法

该方法主要从一些头部标签的知识转移到尾部标签上。例如：在每个类别标签上使用纠错分类模型(Error Correcting Output Codes，ECOC)，该模型依次给每个类别训练一个二进制的分类器。给定的每个类别标签可以使用该类别的自我原始特征和前一个类别标签分类器的预测共同训练分类器。同样我们也可以利用深度极端多标签学习模型(Deep Extreme Multi-label Learning，DeepXML)在头部类别标签和尾部类别标签上同时训练，接着将头部标签的语义表示转移到尾部标签模型中，这样也可以提升模型对底层长尾标签的预测性能。

基于现阶段对于多标签文本分类任务的问题，结合研究者们所提出的模型优化方向，文中提出了一种基于局部法和全局法相结合的方法，即在局部利用不同卷积核的大小结合自注意力机制提取文本信息中的不同维度的数据特征，并结合了全局的方法。该方法利用标签信息的上下位层级关系或标签的层次结构信息对文本的数据特征关系进行建模。然后将所获得的数据特征进行相似度计算，结合 Softmax 将相似度矩阵进行归一化处理，利用阈值筛选。接着对具有上下位

关系的标签建立树形结构关系，得到层次结构标签树。对于以上构建表示层次结构的标签表示方法，现阶段常用的方法有两种：利用树形结构表示法，以及利用有向无环图结构表示法。

针对上述描述，本文提出了基于并行卷积网络信息融合的层级多标签文本分类算法，目的是实现层次多标签文本分类的任务。本文的主要工作内容如下：

1）利用 BERT-Base 预训练模型对文本进行词嵌入，从 3 个维度获取文本信息的数据特征：字（Token）嵌入、句子（Segment）嵌入与位置（Position）嵌入。将 3 个对应嵌入表示进行元素求和，得到两个形状为（1，n，768）的词向量矩阵与字向量矩阵表示。

2）将获得的词向量矩阵分为 CLS-Features 矩阵与 Another-Features 矩阵，将 Another-Features 矩阵输入基于自注意力机制改进的 Parallel-TextCNN 模块。通过使用不同的卷积核大小，来抽取文本信息间不同深度的语义特征。

3）Parallel-TextCNN 模块分为 TextCNN-TopLayer 与 TextCNN-LowLayer。它们都使用 Another-Features 矩阵作为其输入。TextCNN-TopLayer 和 TextCNN-LowLayer 将使用不同的卷积核对输入进行不同深度的数据特征提取。

4）经过并行的 Parallel-TextCNN 模块以及池化层的最大池化（max-pooling）操作后利用 dropout 操作防止数据过拟合，最终将其输入到连接层。在全连接层经过 softmax 归一化处理后，将利用字典保存每一个模块对应不同层的标签概率情况以及对应的层级标签类别与输入到 softmax 函数前的数据特征矩阵。

5）利用数据特征矩阵拼接 CLS-Features 矩阵，恢复原本带有 CLS 的文本语义特征矩阵，将数据作为 Similar-Tree 模块的输入。然后利用其标签信息词向量与恢复的文本语义特征矩阵进行相似度计算，并利用 softmax 进行相似度归一化处理，将归一化后的相似度使用阈值筛选建树的规则进行上下位标签建树。最终输出权重最大的树根节点与其左子节点作为预测的层级标签。

在文本数据中通过将上述局部法和全局法相结合，使用文本数据特征信息与层级标签上下位的依赖关系，利用树形结构充分建模层级依赖。本文实现了层级多标签文本分类算法，该算法主要将上层特征表示中的关键信息传播到下层特征的表示中，然后利用上层任务来指导下层进行预测任务，实验目的是为了提高底层标签的预测性能，同时降低层级标签预测的不一致性。

2 基于并行卷积的深度神经网络模型

基于 Self-Attention 改进的 BPTCNN 模型结构如图 1 所示，该模型主要由 BERT 词嵌入模块、Parallel-TextCNN 并行卷积模块以及 Similar-Tree 树形结构对层级标签进行上下位关系构建模块组成。

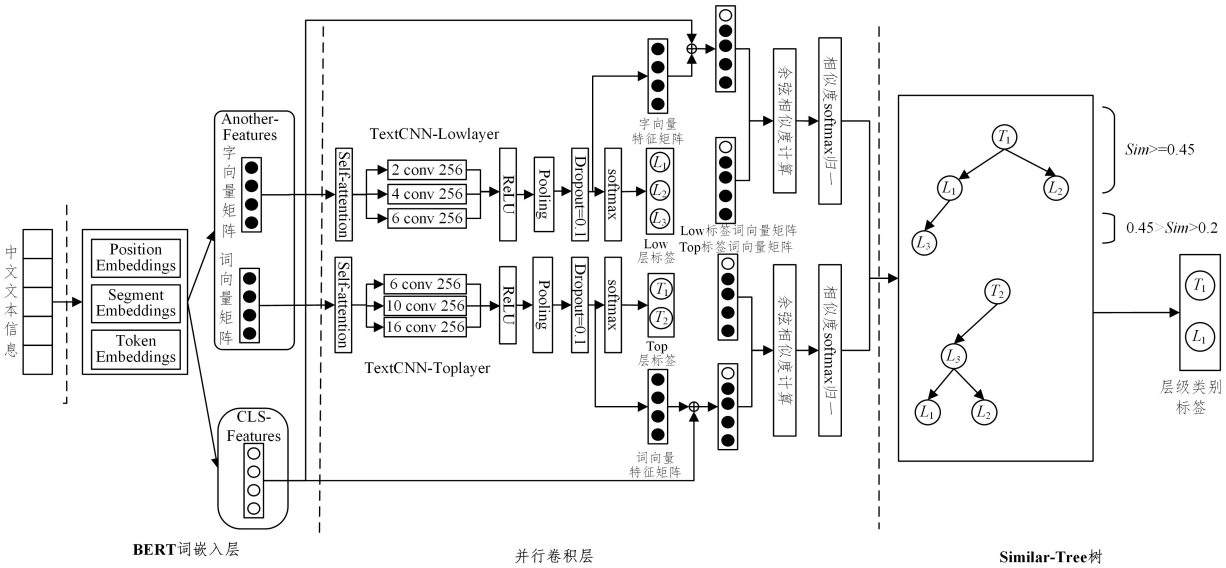


图 1 基于 Self-Attention 改进的 BPTCNN 模型的结构

Fig. 1 Structure of the improved BPTCNN model based on Self-Attention

2.1 BERT 预训练模型动态词向量与字向量

首先，对于中文的文本数据，需要进行词向量编码，将中文文本编码成计算机可以理解的数据特征。接着，利用 BERT 预训练模型和双向 Transformer 编码器，并以遮蔽语言建模（Masked Language Model，MLM）和下一句话预测（Next Sentence Prediction，NSP）为无监督目标，让模型输出的每个字与词的向量表示都更加全面准确地刻画输入文本的整体信息，并最终获得对应词向量与字向量的表示形式。

BERT 模型下游训练词向量时，输入的将不再是单一文本数据，而是将输入文本数据的每一个词（Token）送入其 Token Embedding 层从而获得的一个每一次转换的动态词向量。对于每一个 Token，它的表征有对应的词表征（Token Embedding）、句子表征（Segment Embedding）和位置表征（Position Embedding），其中每一个 Embedding 的维度都是（1，n，768）维，将这对应的 3 个 Embedding 按元素相加，会得到一个大小为（1，n，768）的合成表示，此时便获得了 BERT 编码层的输入。BERT 编码层的模型架构如图 2 所示。

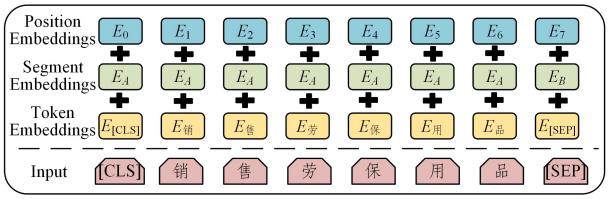


图 2　BERT 模型的 Embeddings

Fig. 2　Embeddings of BERT model

BERT 模型将对应编码层的输出传递到神经网络中，同时神经网络获取其中的隐藏状态，并将隐藏状态全部存储在对象“encoded_layers”中，这将获得 4 个维度的信息：深度神经网络的层数（12 层）、batch 号（句子的个数）、单词/令牌号（每一句中的单词个数）、隐藏单元/特征号（768 个特征）。接着，从上述隐藏状态构建 BERT 文本输入的词向量与字向量。想要获得对应的词向量，由上述隐藏状态每一个词可得到 12 个长度为 768 的单独向量。为了得到单一词向量更好的数据表示，将组合一些层向量。经过实验发现，利用最后 4 层层向量组合可以获得最优的数据特征表示。对于词汇表之外的单词，由于是多个句子和字符级嵌入组成的词汇表之外的单词，平均嵌入方法是最优选择。

BERT 模型架构是一种多层双向变换器（Transformer）编码器，对应结构如图 3 所示。

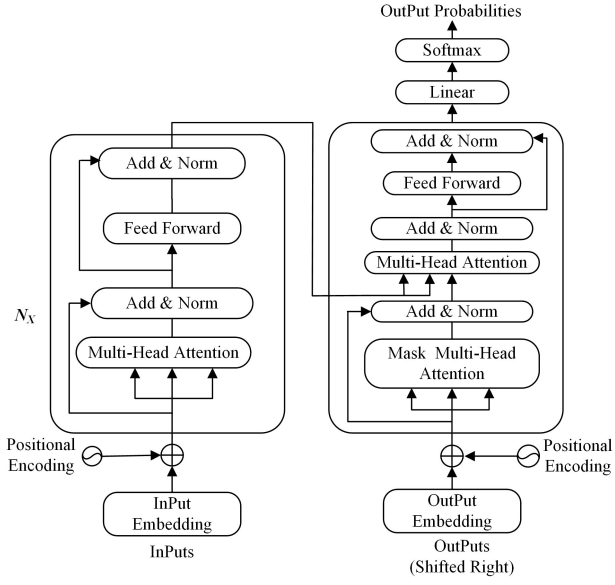


图 3　BERT 的双向 Transformer 结构

Fig. 3　BERT's bidirectional Transformer structure

在 Transformer 的 encoder 中，首先利用自注意力机制（Self-Attention）获得一个加权之后的特征向量 $\boldsymbol{Z}$，它便是注意力 Attention 的计算结果。其计算公式如下所示：

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{1}$$

其中，$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 代表词向量的 3 个矩阵表示，d_k 表示其向量的维度。

其次，利用 Transformer 结构捕捉其位置顺序序列，引入位置编码（Position Embedding）。其对应的公式如下所示：

$$PE(pos,2i)=\sin\left(\frac{pos}{10\,000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{2}$$

$$PE(pos,2i+1)=\cos\left(\frac{pos}{10\,000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{3}$$

其中，pos 表示单词的位置，即 Token 在序列中的位置号码，且 $pos\in[0,maxquelength]$，$maxquelength$ 表示单词序列的最大长度；位置向量的维度用 d_{model} 表示，且与此时整个模型的隐藏状态维度值相同；i 表示单词的维度，它的取值范围是 $[0-d_{\mathrm{model}}/2-1]$ 且为整数；$2i$ 表示向量维度中的偶数维，$2i+1$ 表示向量维度中的奇数维；PE 则是一个行数为最大序列长度 $maxquelength$、列数为 d_{model} 的一个矩阵，即此时该矩阵的形状为 $[maxquelength,d_{\mathrm{model}}]$。利用这样的位置编码结构，后续在模型训练过程中可以使梯度下降更快。其次，在增加位置编码的过程中引入了周期性函数，这样就可以利用周期来表示词对之间的相对位置关系。

BERT 词嵌入中的 Segment Embedding 用于保存文本信息中的句对关系。假如输入的文本中有两句话且有先后顺序关系，BERT 模型将记录该句子向量属于第一句还是第二句，两种形式通过 0，1 来进行标记，该标记序列将记录在 Segment Embedding 中。最后，基于 BERT 预训练模型，该句话的 Segment Embedding 的形状为 $(2,n,768)$，2 表示句子标签的个数，768 表示 BERT 预训练模型的 768 维的向量维度，n 表示长度为 n 的输入序列所获得的 3 种不同的向量表示。

在 BERT 模型词嵌入的 Token Embedding 中，则是将每个字向量转换为固定的维度向量。首先将文本的输入进行 Tokenization 处理，并将每一个字转化为一个 768 维的 WordPiece Token 向量，后续生成的词向量的 TokenEmbedding 将每一个字的 WordPiece Token 进行拼接，组成最后的词张量或词矩阵，即 Token Embedding。

基于 BERT 模型的词嵌入，我们将利用 Token Embedding，Segment Embedding 和 Position Embedding 进行合成表示形成对应文本的词向量 $\boldsymbol{s}_i$ 表示以及字向量 $\boldsymbol{t}_i$ 表示。输入长度为 n 的文本序列所获得的 3 种 Embedding 中，$tok\text{-}emb$ 指代 Token Embedding，$seg\text{-}emb$ 指代 Segment Embedding，$pos\text{-}emb$ 指代 Position Embedding，且这 3 个 Embedding 的形状均为 $(1,n,768)$，1 表示输入序列为 1 个，n 表示文本序列的长度，768 表示 BERT 预训练的向量维度。我们还将利用 Segment Embedding 与 Token Embedding 来进行特征融合，并利用 Layer Norm 将 BERT 模型中的 Embedding 结构统一到相同的分布中，从而得到最终的 $(1,n,768)$ 的合成表示的词向量。而字向量则是利用矩阵保留文本序列中的 Token Embedding 向量。基于 BERT 的预训练模型利用一个长度为 n 的文本序列作为输入，该模型生成对应的词向量与字向量的特征表示的公式如下所示：

$$\boldsymbol{s}_i=LayerNorm(tok\text{-}emb+seg\text{-}emb+pos\text{-}emb) \tag{4}$$

$$\boldsymbol{t}_i=tok-emb \tag{5}$$

结合上述 BERT 模型的 Token，本文将使用哈工大讯飞实验室提供的中文预训练模型 bert-base-chinese 与英文预训练模型 bert-base-cased 对后续文本进行预训练。词向量的维度与字向量的 hidden-size 为 768 维，句子 pad-size 设置为 18，批处理的 batch 为 128，每一句话的学习率为 2×10^{-6}。在

以下公式中，x 表示输入文档文本中所包含的词的个数，n 表示文档中包含有 n 个字，其中每一个向量的维度均为 768。基于文本预训练模型所获得的词向量矩阵 $\boldsymbol{S}$、字向量矩阵 $\boldsymbol{T}$ 的表示形式为：

$$\boldsymbol{S}=[s_1,s_2,\cdots,s_x]^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{T}=[t_1,t_2,\cdots,t_n]^{\mathrm{T}} \tag{7}$$

2.2 并行卷积网络层

经过编码层后将获得对应的词向量矩阵与字向量矩阵，将其内部的词向量与字向量作为并行卷积网络层的输入并对其进行文本数据的特征提取。为了充分提取对应文本数据的特征信息，本模型增加了自注意力机制的并行 TextCNN 对其文本内容信息进行提取。本文提出的并行卷积网络层，将对字向量矩阵抽取低阶信息的模块称为 Textcnn-Lowlayer 模块，将句子中的各个词向量矩阵组成的文本信息抽取高阶信息的模块称为 Textcnn-Toplayer 模块。

2.2.1 Textcnn-Lowlayer 模块

本文使用增加了自注意力机制的 TextCNN 模型在卷积层设置不同的卷积核大小，以应对高阶信息和低阶信息的特征提取。TextCNN 模型主要由卷积层、池化层、非线性激活层、dropout 操作以及全连接层组成。其中在卷积层引入的自注意力机制，其输入是 BERT 模型提供的字向量矩阵的 Another-Features。

1)卷积层

引入自注意力机制，是为了从字向量矩阵筛选出少量重要信息并增加某些关键词的权重以提取文本中重要的语义特征，利用字向量在卷积层中卷积核的大小对文本数据进行低阶特征提取。在 Textcnn-Lowlayer 模块，采用大小为 2，4 和 6 的卷积核对低阶数据进行局部特征词采取，其卷积过程表示为：

$$h_i=f\left(\sum_{x=1}^{3}\sum_{y=1}^{3}w_{i(x,y)}\times c_{(x,y)}+b_i\right) \tag{8}$$

其中，h_i 代表卷积层的结果；f 代表激活函数，该模型采用的是 ReLU 激活函数；$w_{i(x,y)}$ 表示输出矩阵中第 i 个节点对应过滤器输入节点 (x,y) 的权重；b_i 是其第 i 个节点的偏置项；$c_{(x,y)}$ 表示过滤器中节点 (x,y) 的值。由于共享权重，此时 w 和 b 在卷积核中均相同。对于自注意力机制而言，所有的 $\boldsymbol{Q}$，$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 均来自于自身的词，故 $\boldsymbol{Q}=\boldsymbol{K}=\boldsymbol{V}$，此时编码层中每一个位置都会处理编码器前一层的输出。

2)池化层

池化层可以让模型更加注重某些特征，同时也能通过缩减特征向量和网络参数的大小达到降维的目的。本文对卷积操作的输出 h_i 进行池化处理，利用最大池化方法实现。池化层的输出将作为融合层的输入，融合层的目的是将 3 个池化层所提取的特征进行拼接，并形成一个具有代表性的组合特征向量。

3)非线性激活层

利用非线性激活层，可以使神经网络的学习能力得到强化。本模型使用 ReLU 函数进行激活，x 代表其对应输入激活函数的值。对应 ReLU 激活函数的公式为：

$$f(x)=\max(0,x) \tag{9}$$

4)dropout 操作

dropout 操作是池化层的输出进入全连接层前的一步操作。使用 dropout 操作是为了防止数据过拟合，并丢弃无效特征数据，降低后续无效特征数据对模型的影响。

5)全连接层

全连接层在 Textcnn-Lowlayer 模块中分为两层：第一层使用 ReLU 函数进行激活；第二层使用 softmax 函数将对应的 low 标签进行概率的归一化计算，此时需要使用字典保存 softmax 函数输出的所有概率最大的前 3 个类别与其概率值，并保存利用 ReLU 函数激活的数据特征矩阵。

2.2.2 Textcnn-Toplayer 模块

Textcnn-Toplayer 模块的输入是词向量矩阵对应的 Another-Features 矩阵。在卷积层上，与 Textcnn-Lowlayer 模块的卷积核大小上有所不同，Textcnn-Toplayer 模块中设置的 3 个卷积核大小为 6，10 和 16。在全连接层的第二层，利用字典保存 softmax 函数结果的 top 标签概率最大的前两个标签及其概率值。

2.3 输出层

经过 Textcnn-Toplayer 模块与 Textcnn-Lowlayer 模块对文本数据特征的提取，将获得对应层级标签中 top 层标签中 2 个概率最大值标签和 low 层标签中 3 个概率最大值标签。由于该分类任务最终目的是要输出一对具有层级关系的标签对，因此本文的输出层增加了阈值筛选模块与 Similar-Tree 模块。

2.3.1 层级标签相似度计算

利用全连接层，经过 ReLU 函数的特征矩阵与 BERT 模型的对应 CLS-Features 进行拼接操作。我们将 top 层标签的数据特征矩阵与词向量矩阵对应的 CLS-Features 相拼接，low 层标签的数据特征矩阵将与字向量矩阵对应的 CLS-Features 相拼接，拼接的目的是恢复其原有语义信息，增强标签之间的语义关系。

对应的字向量拼接公式为：

$$\boldsymbol{T}=t_G^l\oplus t_G^s \tag{10}$$

对应词向量的拼接公式为：

$$\boldsymbol{W}=w_G^l\oplus w_G^s \tag{11}$$

其中，$\oplus$ 为拼接操作，$\boldsymbol{W}$ 表示拼接后恢复原语义的词向量数据特征矩阵，$\boldsymbol{T}$ 表示拼接后恢复原语义的字向量数据特征矩阵，t_G^s 代表字向量的 CLS-Feature，w_G^s 代表词向量的 CLS-Feature。

对于 top 层标签与 low 层标签，将利用 BERT 词向量模型输出其对应的词向量表示。将对应 top 层标签的词向量与恢复语义的词向量数据特征矩阵按权重进行聚合。经过多次实验，发现设置标签权重 $w_{\mathrm{lab}}=0.43$，词向量矩阵权重 $w_{\mathrm{wd}}=0.57$，可以使最终向量表达形式数据最容易发现对应层级关系。此时按权重聚合的公式为：

$$cbmat_i=w_{\mathrm{lab}}\cdot labemb_i+w_{\mathrm{wd}}\cdot wdemb_i \tag{12}$$

利用上述公式完成对 top 标签对应的词向量矩阵、low 标签对应的字向量矩阵的加权计算，得到恢复其对应文本语义的 top 标签加权矩阵和 low 标签加权矩阵。

然后，利用构建的 top 标签加权矩阵与 low 标签的加权矩阵，衡量对应矩阵中每个向量的相似性。此处，引入余弦相似度计算公式作为其向量的相似度量。余弦相似度计算公式为：

$$\cos(\theta)=\frac{\sum_{i=1}^{N}\alpha_i b_i}{\sqrt{\sum_{i=1}^{N}\alpha_i^2}\sqrt{\sum_{i=1}^{N}b_i^2}} \tag{13}$$

利用上述公式计算每一个 top 标签与 low 标签对应的相似度，并使用 softmax 函数对其进行归一化处理，同时用字典保存其归一化后的 low 标签和 low 标签对应 top 标签的相似度概率。

2.3.2　利用阈值筛选建树

经过多次实验，发现当 top 标签与 low 标签在相似度概率归一化后其相似概率大于 0.45 时，表明 top 标签与 low 标签是具有层级关系的标签对；当相似度概率大于 0.2 但小于 0.45 时，表明 top 标签与 low 标签可能存在一定的上下位层级关系。

故可以利用上述归一化后的相似度阈值结合实体类型的层次结构构建具有上下位关系的树。当树中子类有效时，该子类所属的父类一定有效，因此我们可从 low 标签中找到对应的具有层级关系的 top 标签，反之亦然。在本文的树结构中，左子节点的相似度值大于其右子节点，即可查找到其对应层级标签对的父类与子类的映射关系。Similar-Tree 模块的结构如图 4 所示。

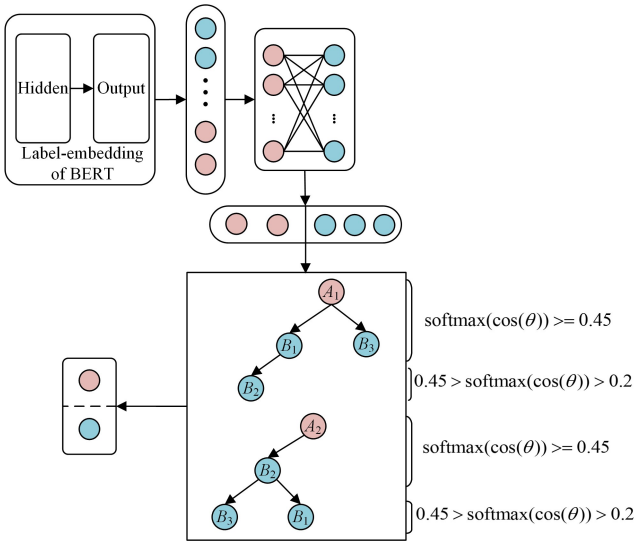


图 4　Similar-Tree 模块

Fig. 4　Module of Similar-Tree

当所有 top 标签与 low 标签的树建立完后，将选择其第一层相似度权重概率最大的树进行节点的输出，输出其根节点作为 top 标签，其根节点的左子节点作为 low 标签。若此时根节点的第一层存在右子节点，则将其选为候选 low 标签进行保留（需要两个下级标签时进行保留）。

3　实验设置

3.1　实验环境

本节通过实验对本文算法的可能性与有效性进行验证，实验环境的配置信息如表 1 所列。

表 1　实验环境配置

Table 1　Experimental environment configuration

Lab Environment	Specific Information
Operating System	Ubuntu 18.04
CPU	AMD Ryzen 5 3600
GPU	NVIDIA RTX 3080
Development Language	Python 3.7.1
Development Platform	Pytorch 1.6.0

3.2　实验数据

本实验选用了具有多标签的公开 Kanshan-Cup 数据集与非公开数据集 CI 企业信息数据集，以充分检验算法的性能。

1）Kanshan-Cup 数据集

Kanshan-Cup 数据集由知乎 2017 年看山杯比赛所提供。该数据集是一个文本多标签分类问题，训练数据给出了 300 万个问题及其话题的绑定关系，话题标签有 1 999 个，对应话题之间存在父子层级关系，所以其也是一个典型的层级多标签文本分类数据集。由于该数据集中标签类别数太多，故本实验选取 10 个标签，即 5 对具有上下层级关系的标签。在每个问题类别数据集中取 1500 条构建训练数据集，取 200 条构建验证集，另取 200 条构建测试集。由于数据进行了脱敏处理，我们使用类别标签码作为其标签。

2）CI 企业信息数据集

CI 企业数据集中的类别标签数据由国家对应的行业信息分类体系构建，包含国家一级行业 19 个类别，对应 90 个二级行业类别。其中，一级和二级行业类别具有上下位层级关系，主体信息则对应公司的主要经营范围信息。该数据集共有 87000 条数据，其中包含一级行业类别 5 个和其对应二级行业类别 51 个。对于每个子类别，选取 300 个对应公司的经营信息，共 15300 条作为训练集，选取 100 条作为验证集，100 条作为测试集。CI 企业信息数据样本示例如表 2 所列。

表 2　CI 企业信息数据样本

Table 2　Samples of CI enterprise information data

Textcontent	First Class Label	Second Class Label
生态农业开发，农业产品	农、林、牧、渔业	农业
毛竹，林木种植，苗木培育	农、林、牧、渔业	林业
矿产品收购，销售	采矿业	其他采矿业

3.3　对比模型与参数设置

为了检验本文算法相比传统层级多标签文本分类模型在性能上的优势及其对层级多标签文本分类预测性能的提升，将其与已有模型进行对比实验。参与对比的模型主要分为两类：一类是基于深度学习的 TextCNN，TextRNN，FastText，BiGRU 以及 RCNN 模型；第二类则是基于现阶段主流的图卷积的 TextGCN 模型以及 GAT 模型。

本实验的主要参数设置如表 3 所列。在 Kanshan-Cup 数据集上，pad_size 值设置为 32。在 CI 企业信息数据集上，pad_size 值设置为 18。为了降低模型过拟合的风险，将设置模型连续 1000 个 batch 的训练无明显提升时，就提前终止模型的训练。表 3 分割线以上展示的是部分深度学习模型的参数，分割线以下展示的是部分图卷积的 TextGCN 模型的参数与

GAT 模型的部分参数，并使用 L2 正则化和 Adam 优化器对网络进行优化。

表 3 主要实验参数

Table 3 Main experimental parameters

Parameter	Value
batch_size	128
learning_rate	2×10^{-6}
dropout	0.1
Toplayer_filter_sizes	6,10,16
Lowlayer-filter_sizes	2,4,6
TextCNN_filter_sizes	2,4,6
hidden_size	768
epoch	5
lr	0.02
dropout	0.5
weight_decay	0.01
alpha	0.2
weight_decay	5e-3
hidden	8

3.4 评价指标

本实验采用宏准确率(MacroP)、宏召回率(MacroR)和综合指标微 F1 值来衡量层级多标签文本分类模型的性能。

$$marcoPrecision=\frac{1}{n}\sum_{i=1}^{N}P_i \tag{14}$$

$$marcoRecall=\frac{1}{n}\sum_{i=1}^{N}R_i \tag{15}$$

$$microRecall=\frac{\sum_{1}^{n}TP_i}{\sum_{1}^{n}TP_i+\sum_{1}^{n}FN_i} \tag{16}$$

$$microPrecision=\frac{\sum_{1}^{n}TP_i}{\sum_{1}^{n}TP_i+\sum_{1}^{n}FP_i} \tag{17}$$

$$micro\text{-}F1=2\frac{microRecall\times microPrecision}{microRecall+microPrecision} \tag{18}$$

由于此时 *micro-F1* 是微召回率(*microRecall*)和微准确率(*microPrecision*)的调和平均，因此这 3 个值大小相等。

4 实验结果分析

为了综合评估本文算法与常用文本分类模型对层级多标签文本分类识别的优劣，分别在 Kanshan-Cup 数据集和 CI 企业信息数据集上进行实验，同时选用宏准确率、宏召回率和微 F1 值作为其评价指标。

1)在 Kanshan-Cup 数据集上的结果分析

在 Kanshan-Cup 数据集上进行实验，选用 3.3 节所提及的模型作为对比，实验结果如表 4 所列。

表 4 在 Kanshan-Cup 数据集上的实验结果

Table 4 Experimental results on Kanshan-Cup dataset

Algorithm	*MacroP*	*MacroR*	*MicroF1*
Word2vec+TextCNN	35.40	37.60	38.40
Word2vec+TextRNN	39.67	40.87	39.26
Word2vec+FastText	38.64	34.72	36.87
Word2vec+BiGRU	40.07	39.94	39.97
Word2vec+RCNN	39.89	39.77	39.46
Word2vec+TextGCN	53.47	53.83	52.76
Word2vec+GAT	57.75	55.44	56.91
Self-attention+BPTCNN	**58.95**	**59.77**	**57.63**

由表 4 可知，在 Kanshan-Cup 数据集上，本文算法在 3 类评价指标上均优于对比模型。在综合性评价指标上，本模型相比基于深度学习的 TextCNN，TextRNN，FastText，BiGRU，RCNN 模型和基于图卷积模型的 TextGCN 与 GAT 模型，分别提升了 19.23%，18.37%，20.76%，17.66% 和 18.17%。

然而，相比现阶段主流的利用图卷积的模型 TextGCN 以及 GAT，本文模型的提升则比较有限，其相比 TextGCN 模型提升了 4.48%，相比 GAT 模型提升了 2.08%。

从微 F1 值的提升幅度可以看出，采用 self-attention 和改进的 BPTCNN 模型在准确率上的提升均优于使用单一文本分类模型在多标签文本分类模型。这也可以说明利用 BERT 预训练模型作为输入比使用 Word2Vec 模型作为输入更能获得对应词和句子的动态词向量，动态词向量模型基于上下文词的考虑能更好地表示其语义。该项对比彰显了利用预训练模型的文本表示能力优于传统的静态词嵌入模型。因此，基于对 BERT 预训练模型生成的动态字向量和动态词向量维度的双层数据表示，能够提升模型对其数据特征的提取能力和学习能力，从而有助于多标签文本特征的提取。

从 MacroP 和 MacroR 两项评价指标来分析，BPTCNN 模型在各标签类别上的宏准确率和宏召回率均有较大提升。优化后的模型相比于深度学习类的模型，在 MacroR 指标上最多提升了 25.05%(为 BPTCNN 模型与 FastText 模型)，最少提升了 18.9%(为 BPTCNN 模型与 TextRNN 模型)；在 MacroP 指标上最多提升了 23.55%(为 BPTCNN 模型与 TextCNN 模型)，最少提升了 18.88%(为 BPTCNN 模型与 BiGRU 模型)。

上述对比结果也验证了优化后的模型利用标签上下位层级关系来增强其标签间的上下位联系，使得上下位关系之间可以用标签对的形式来实现层级关系的构建。相比基于图卷积的 TextGCN 模型与 GAT 模型，优化后的模型在 MacroP 指标上最多提升了 5.48%(为 BPTCNN 模型与 TextGCN 模型)，最少提升了 1.2%(为 BPTCNN 模型与 GAT 模型)；在 MacroR 指标上最多提升了 5.94%(为 BPTCNN 模型与 TextGCN 模型)，最少提升了 4.33%(为 BPTCNN 模型与 GAT 模型)。

因为现有的图卷积模型将抽取的文本数据作为图中的节点，将数据间的上下位关系作为连接节点的边，所以图卷积模型在对应层级标签的上下位构建中也比较有效。该项对比实验也证明了本文模型在上下位关系抽取方面较为有效，在利用上下位关系辅助标签节点进行层级关系的构建上是有效的。

为了验证所提算法在对应标签类别上的分类效果，引入精度(Precision@K)与归一化折扣累计收益(Normalized Discounted Cumulative Gain@K)。

$$P@K=\frac{1}{k}\sum_{L\in rankk(\hat{y})}^{k}y^L \tag{19}$$

$$DCG@K=\sum_{L\in rankk(\hat{y})}^{k}\frac{y^L}{\log(L+1)} \tag{20}$$

$$NDCG@K=\frac{DCG@K}{\sum_{L=1}^{\min(k,|y|_0)}\frac{1}{\log(L+1)}} \tag{21}$$

其中，k 为模型预测结果中的位置索引，$\boldsymbol{y}\in\{0,1\}^L$ 是输入

文本所对应的真实标签向量，$\hat{\boldsymbol{y}} \in R^L$ 是模型预测的标签向量。

各个模型在 Kanshan-Cup 数据集上基于评价指标 P@K 和 NDCG@K 的对比情况如图 5 所示。

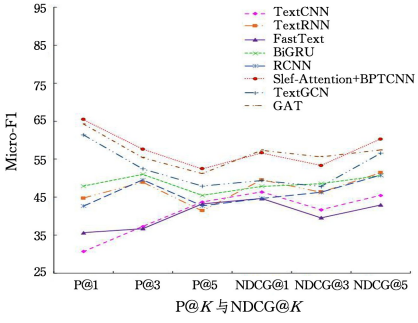


图 5　各模型算法在 P@K 与 NDCG@K 上的对比

Fig. 5　Comparison of each model algorithm in P@K and NDCG@K

由实验对比数据可以看出，本文算法在具有层次关系的多标签文本分类数据上相比其余深度学习算法有较好的提升，在父类标签拥有一个子类时，其准确率提升最为明显。当一个父类标签有多个子类标签时，其准确率提升较有限。因为一个父类拥有多个子类时，可能存在各个子类之间的强相关性，导致父类寻找子类时难以确定层级关系，从而导致分类难度增大。

从图 5 中也可以看出，BPTCNN 模型在多类指标上均比深度学习模型更优，在预测性能上获得了较大的提升。但对比图卷积模型，BPTCNN 模型的性能在 NDCG@1 与 NDCG@3 上略逊色于 GAT 模型，这是由于 BPTCNN 模型在数据集中存在部分的父类与对应子类在建立阈值筛树时未达到建树条件，所以在构建上下位层级关系树时未能找到该父类与对应子类对应的上下位关系；而 GAT 模型在构建边的关系时仅利用节点的数据，模型可能获取到一些本文模型的阈值筛选建树不成功的节点关系。虽然在部分数据中，本文模型的表现略差于现有的 GAT 模型，但在整体预测性能上本文模型有更优的表现。

2）在 CI 企业信息数据集上的结果分析

在 CI 企业信息数据集上，选用 TextCNN，TextRNN 和 BiGRU 深度学习模型以及 TextGCN 和 GAT 图卷积模型作为对比模型，实验结果如表 5 所列。

表 5　在 CI 企业信息数据集上的实验结果

Table 5　Experimental results on CI Enterprise datasets

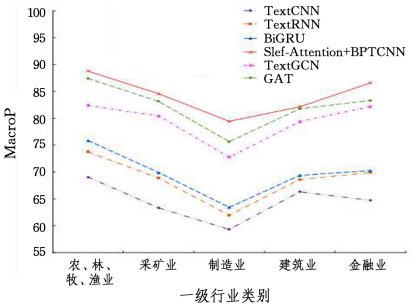
Algorithm	*MacroP*	*MacroR*	*MicroF*1
Word2Vec＋TextCNN	64.57	62.79	63.45
Word2Vec＋TextRNN	68.62	68.43	68.56
Word2Vec＋BiGU	69.79	70.02	69.94
Word2Vec＋TextGCN	79.41	76.77	78.63
Word2Vec＋GAT	82.37	81.65	82.14
Self-Attention＋BPTCNN	**84.32**	**82.75**	**83.77**

由表 5 可知，在 CI 企业信息数据集上，本文模型在 3 类评价指标上均有较好的提升。

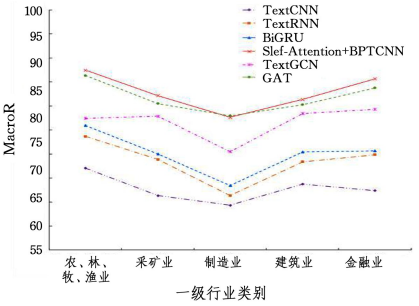
在与深度学习模型对比方面，相比于 TextCNN 模型，本文模型在宏精确率、宏召回率和微 F1 值指标上分别提升了 19.75%，19.96%和 20.32%。相比于 TextRNN 模型，本文算法的宏精确率、宏召回率和微 F1 值分别提升了 15.7%，14.32%和 15.21%。相比于 BiGU 模型，本文算法的宏精确率、宏召回率和微 F1 值分别提升了 14.53%，12.73%和 13.83%。

而对比图卷积模型来说，BPTCNN 模型的性能提升较为有限。因为图卷积模型也利用节点间的关系对上下位信息进行了抽取，所以可以获得部分文本之间的上下位数据特征。相比于 TextGCN 模型，本文模型在宏精确率、宏召回率和微 F1 值指标上分别提升了 4.91%，5.8%和 5.14%，相比于 GAT 模型分别提升了 1.95%，1.1%和 1.63%。

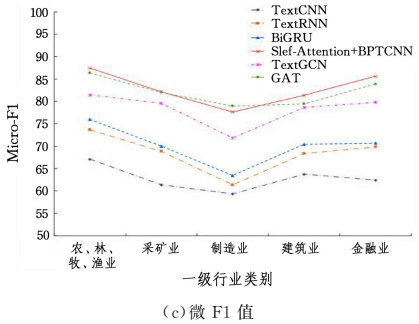
为了对比上述模型在 CI 数据集上每一个一级行业类别标签的分类效果，在 5 个企业类别上进行对比实验，并绘制各个模型在 5 个一级行业下的宏精确率、宏召回率与微 F1 值的结果，如图 6 所示。



(a)宏准确率



(b)宏召回率



(c)微 F1 值

图 6　各个模型在 5 个一级行业数据上的分类效果

Fig. 6　Classification effect of each model on 5 first-level industry data

由图 6(a)可知，本文算法在 CI 企业数据集上的宏准确率均高于对比模型，特别是在宏精确率普遍最低的制造业类中，取得了 4 类模型中的最高分值。在宏召回率评价指标上，由图 6(b)可知，本文算法在宏召回率普遍较低的金融业上的成绩相比 3 类深度学习模型获得了较大的提升。由图 6(c)可知，在微 F1 值上，本文算法在 CI 数据集各一级行业下均高于 78.4，本文算法的总体表现情况优于所对比的其他 3 类

模型。实验结果表明本文算法对层级多标签文本分类的效果优于单一深度神经网络算法对层级多标签文本分类的效果。

从图 6(c)中可以看到 GAT 模型在制造业的表现比本文模型略有优势，但在其他企业类别的数据中，本文模型较图卷积类模型仍具有一定程度的提升，也说明利用本文模型可以很好地对层级多标签文本分类任务做到较高的识别准确率。

结束语 针对层级多标签文本分类任务存在的难以从单一文本信息中抽取高维和低维数据特征与表示，以及层级多标签文本分类任务存在的难以使用单一数据特征结合标签构建上下位层级标签关系的两类问题，本文提出了基于 Self-Attention 改进的 BPTCNN 的层级多标签文本分类算法。

首先，该算法利用 BERT 预训练模型对输入数据进行字向量与词向量训练。然后，利用增加自注意力机制的并行卷积网络，分别基于不同卷积核大小与不同卷积深度对字向量矩阵与词向量矩阵表示的文本信息进行高维和低维特征提取。接着，将两个神经网络模型提取到的特征恢复原语义特征后，利用相似度结合 softmax 层进行相似度归一化处理，利用树形结构对相似度归一化处理后的上下位标签层级关系进行建树。最后，完成层级多标签文本分类任务。

实验结果表明，本文算法虽然能够在层级多标签文本分类任务上取得良好的效果，但是在预训练模型的词向量获得较好的动态语义的同时需要较优的硬件支持去保存对应的词向量矩阵和字向量矩阵。因此，下一步将考虑使用其他维度较小的词向量且效果良好的预训练模型的词向量与字向量替换基于 BERT 模型所提供的词向量与字向量。

参 考 文 献

[1] WU S, GAO M, XIAO Q, et al. A topic-enhanced recurrent autoencoder model for sentiment analysis of short texts[J]. International Journal of Internet Manufacturing and Services, 2020, 7(4): 393-406.

[2] BIN N, WU J W, HU F. Spam message classification based on the Naïve Bayes classification algorithm[J]. IAENG International Journal of Computer Science, 2019, 46(1): 46-53.

[3] CHEN J, HE J, SHEN Y, et al. End-to-end learning of LDA by mirror-descent back propagation over a deep architecture[J]. arXiv: 1508.03398, 2015.

[4] MINAEE S, KALCHBRENNER N, CAMBRIA E, et al. Deep learning--based text classification: a comprehensive review[J]. ACM Computing Surveys(CSUR), 2021, 54(3): 1-40.

[5] TAN C. Short Text Classification Based on LDA and SVM [J]. International Journal of Applied Mathematics & Stats, 2013, 51(22): 205-214.

[6] YIN C, SHI L, WANG J. Short Text Classification Technology Based on KNN+Hierarchy SVM [C] // International Conference on Multimedia and Ubiquitous Engineering International Conference on Future Information Technology. 2017: 633-639.

[7] JIANG T, WANG D, SUN L, et al. Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification[C] // Proceedings of the AAAI Conference on Artificial Intelligence. 2021: 7987-7994.

[8] JOHNSON R, ZHANG T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv: 1412.1058, 2014.

[9] GARGIULO F, SILVESTRI S, CIAMPI M, et al. Deep neural network for hierarchical extreme multi-label text classification [J]. Applied Soft Computing, 2019, 79: 125-138.

[10] LIU J, CHANG W C, WU Y, et al. Deep learning for extreme multi-label text classification [C] // Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017: 115-124.

[11] KIM Y. Convolutional Neural Networks for Sentence Classification [C] // Proceedings of the Conference on Empirical Methods in Natural Language Processing. 2014: 1746-1751.

[12] GRAVES A, MOHAMED A, HINTON G. Speech recognition with deep recurrent neural networks[C] // Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. 2013: 6645-6649.

[13] JOULIN A, GRAVE E, BOJANOWSKI P, et al. FastText. zip: Compressing text classification models[J]. arXiv: 1612.03651, 2016.

[14] GARGIULO F, SILVESTRI S, CIAMPI M, et al. Deep neural network for hierarchical extreme multi-label text classification [J]. Applied Soft Computing, 2019, 79: 125-138.

[15] ZHENG C, HONG T T, XUE M Y. BLSTM_MLPCNN Model For short Text Classification [J]. Computer Science, 2019, 46(6): 206-211.

[16] DUAN D D, TANG J S, WEN Y, et al. Chinese short text classification algorithm based on BERT model[J]. Computer engineering, 2021, 47(1): 79-86.

[17] LAN Z, CHEN M, GOODMAN S, et al. Albert: A lite bert for self-supervised learning of language representations[J]. arXiv: 1909.11942, 2019.

[18] GARGIULO F, SILVESTRI S, CIAMPI M, et al. Deep neural network for hierarchical extreme multi-label text classification [J]. Applied Soft Computing, 2019, 79: 125-138.

[19] SOUCY P, MINEAU G W. A simple KNN algorithm for text categorization[C] // Proceedings 2001 IEEE International Conference on Data Mining. IEEE, 2001: 647-648.

[20] CAI L, HOFMANN T. Hierarchical document categorization with support vector machines[C] // Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management. 2004: 78-87.

YI Liu, born in 1995, postgraduate. His main research interests include natural language processing and text classification.

GENG Xinyu, born in 1964, professor. His main research interests include data mining and artificial neural networks.

（责任编辑：柯颖）