doi:10.6041/j.issn.1000-1298.2021.S0.023

基于注意力池化和堆叠式结构的病虫害文献识别模型

唐　詹　柏　召　刁　磊　郭旭超　周　晗　李　林
（中国农业大学信息与电气工程学院，北京 100083）

摘要：为解决病虫害文献识别过程中存在语义特征学习不够、上下文信息不能充分利用等问题，以病虫害相关文献摘要为研究对象，提出一种基于注意力池化策略和堆叠式双向长短期记忆（Bi-directional long-short term memory，BiLSTM）的神经网络模型（AP－LSTM）。该模型采用堆叠式长短期记忆结构，提高了对语义特征的学习能力，在进行堆叠操作时，通过将输入向量与输出向量拼接，进一步加强了对语义信息的表征；然后采用基于注意力机制的池化策略为不同的词分配不同权重，使模型在抓住重点的同时能够充分利用上下文信息。本文在包含 1 439 条正例、1 061 条负例的自标注数据集上进行了实验，所提出的 AP－LSTM 模型在该数据集上的精确率、召回率、F1 值和准确率分别为 92.67%、97.20%、94.88% 和 94.00%，实验结果表明，AP－LSTM 模型能够有效识别病虫害文献。

关键词：病虫害；文献识别；注意力机制；双向长短时记忆网络；长文本分类
中图分类号：TP391.1　　**文献标识码**：A　　**文章编号**：1000-1298(2021)S0-0178-07

Diseases and Pests Articles Identification Model Based on Attention Pooling and Stacked Structure

TANG Zhan　BAI Zhao　DIAO Lei　GUO Xuchao　ZHOU Han　LI Lin
(*College of Information and Electrical Engineering*, *China Agricultural University*, *Beijing* 100083, *China*)

Abstract: Diseases and pests articles identification is an important pre-task of natural language processing in the field of diseases and pests. It is of great significance to develop a fast and accurate method for diseases and pests articles identification. In order to solve the problems of insufficient learning of semantic features and insufficient use of context information in the process of diseases and pests articles identification, a neural network model of attention pooling based bi-directional long-short term memory (AP－LSTM) was proposed, which was based on attention pooling strategy and bi-directional long-short term memory (BiLSTM). The model adopted the stacked LSTM structure, which improved the learning ability of semantic features. In the stacking operation, the input vector and output vector were concatenated to further enhance the representation of semantic information. Then, a pooling strategy based on the attention mechanism was used to assign different weights to different words, so that the model can make full use of context information while grasping the keywords. The experiments were carried out on a self annotated dataset with 2 500 labeled samples, including 1 439 positive cases and 1 061 negative cases. The precision, recall, F1 score, and accuracy of the proposed AP－LSTM model on the dataset were 92.67%, 97.20%, 94.88%, and 94.00%, respectively. The experimental results showed that the proposed AP－LSTM model can effectively identify pest literature.

Key words: diseases and pests; articles identification; attention mechanism; bi-directional long-short term memory network; long text classification

0　引言

近年来，面向病虫害领域的自然语言处理吸引了广大研究者的关注，如本体构建[1]、命名实体识别[2-3]、关系抽取[4]、决策系统[5]等。病虫害相关文献是这些工作的重要研究对象，通常通过搜索引

收稿日期：2021－07－12　修回日期：2021－09－08
基金项目：国家重点研发计划项目(2016YFD0300710)
作者简介：唐詹(1995—)，男，博士生，主要从事自然语言处理与深度学习研究，E-mail：tz_blues@163.com
通信作者：李林(1963—)，女，教授，博士生导师，主要从事视频图像处理研究，E-mail：lilincau@126.com

擎检索相关文献，但检索到文献中的一部分与病虫害并不相关，需要在批量导出检索到的文献之后根据文献摘要人工筛选出与病虫害相关的文献，这一操作耗时费力，人力成本高。

通过摘要判断文献是否与病虫害相关本质上是一个二分类问题，考虑到摘要文本的长度，可以将病虫害文献识别视作一个长文本二分类问题。对于文本分类问题，基于卷积神经网络（Convolutional neural networks，CNN）[6-8] 和循环神经网络（Recurrent neural networks，RNN）[9-11] 的模型应用广泛，注意力机制[12] 也发挥着重要作用。针对中文长文本分类，文献[13]用段落向量表示文本句子，然后通过文本类别计算句子的注意力，再以句子注意力的均方差作为贡献度进行过滤，最后用 CNN 和最大池化进行分类，在新闻类长文本分类问题上取得了较好表现；文献[14]以 Seq2Seq 模型为基础，通过多分类标签之间的共生关系提取词义特征和语义特征，构建了基于多级特征和混合注意力机制的多标签分类算法，在知乎长文本和公开数据集上取得了较好效果；文献[15]针对长文本中的冗余信息问题，提出了具有门控机制的融合模型，并使用注意机制计算单词的上下文向量以导出关键字信息，在新闻类长文本数据集上取得了较好结果。

摘要文本作为面向特定领域的长文本，还面临着其它的一些挑战：术语较多，学术文献摘要通常包含一些领域内的专业术语，这些术语往往具有更深层次的词义信息，这要求文本分类模型具有足够强的词义特征提取能力；结构严谨，表达简明，学术文献摘要需要简明扼要地准确总结出整篇文献的内容，且具有较强的逻辑性，这要求文本分类模型能对长距离的上下文信息进行利用，并在不忽略有用信息的同时抓住重点和关键。

因此，为能够更好地完成病虫害文献识别，本文提出一种基于注意力机制和堆叠式双向长短期记忆（Bi-directional long-short term memory，BiLSTM）的神经网络模型（AP－LSTM），该模型以 BiLSTM 为基本框架，通过堆叠式结构加强对词义特征的提取能力，并结合注意力机制捕捉长距离的上下文信息，有重点地对上下文信息进行表征，以提高模型的分类准确性。

1　材料与方法

1.1　数据集构建

1.1.1　文本数据收集与标注

以“病虫害”为检索关键词，从中国知网（https://www.cnki.net/）收集 2 500 篇文献摘要，然后在病虫害领域专家指导下进行人工标注，与病虫害相关的摘要标注为正例，标签为 1，共 1 439 条；与病虫害无关的标注为负例，标签为 0，共 1 061 条。

1.1.2　分词处理

中文文本的分词处理相对英文来说更为复杂，中文词与词之间没有显式的间隔，考虑到单个汉字表达的信息通常弱于词组，因此本文中采用 jieba 工具（https://github.com/fxsjy/jieba/）对每条摘要文本进行分词。分词之后得到的序列长度分布如图 1 所示。

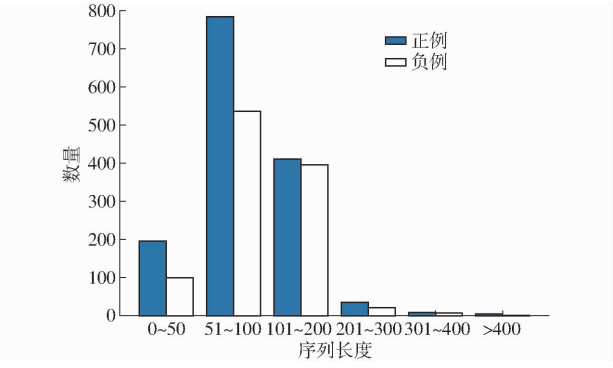

图 1　序列长度分布

Fig. 1　Sequence length distribution

从图 1 中可以看出，经分词之后，大部分摘要文本的序列长度在 51 ~ 200 之间，共 2 127 条，占比 85.08%；序列长度在 51 以下的，共 296 条，占比 11.84%；序列长度超过 200 的，共 77 条，占比 3.08%。

1.1.3　向量化处理

为更加稠密地将文本转换为数值向量，本文采用 Word2Vec[16] 中的 Skip－Gram 模型训练词嵌入向量，词向量维度设置为 100。相比于传统的词表示方法，词嵌入向量不但能从大规模无标签语料中捕获语义信息，也能捕获到句法信息[17]。

1.2　AP－LSTM 模型

针对病虫害相关摘要文本的特点和存在的问题，本文结合注意力机制和堆叠式 BiLSTM，提出了 AP－LSTM 模型，其架构如图 2 所示。

AP－LSTM 模型包括 4 部分：嵌入层用于将输入文本转换成数值向量表示；堆叠层数为 n 的长短期记忆（LSTM）层用于文本建模，⊕表示拼接操作，单个 LSTM 层的操作示意图如图 3 所示；注意力层用于对提取的特征分配权重，实现软性的池化操作；最后使用 Softmax 分类器完成分类。

1.2.1　嵌入层

嵌入层利用训练好的词向量将文本转换为数值向量表示

$$(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_i,\cdots,\boldsymbol{x}_N)=\text{embedding}(w_1,w_2,\cdots,w_i,\cdots,w_N) \quad (1)$$

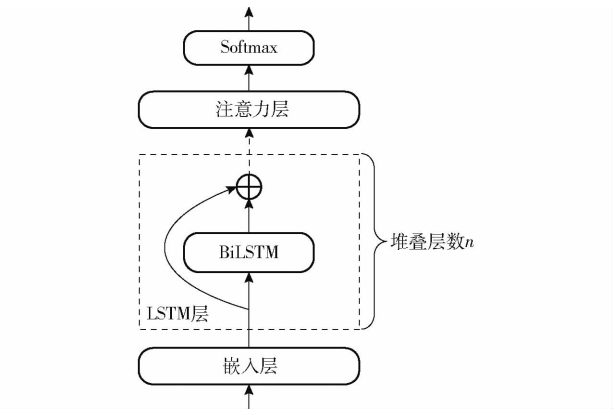

图 2 AP－LSTM 模型架构

Fig. 2 Architecture of AP－LSTM

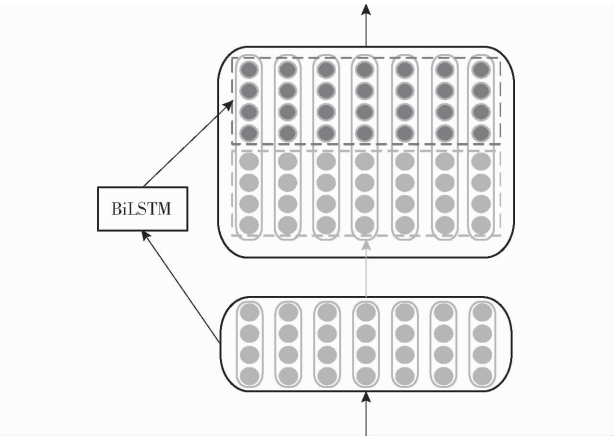

图 3 LSTM 层操作示意图

Fig. 3 Operation schematic of LSTM layer

式中 $(w_1, w_2, \cdots, w_i, \cdots, w_N)$——文本序列中的词

$(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_N)$——$w_i$ 对应的词向量

embedding——嵌入层函数

N——总词数，即摘要文本序列的长度

用 text 表示输入文本，$\boldsymbol{X}$ 表示得到的词向量序列，可以将式(1)简记为

$$\boldsymbol{X} = \text{embedding}(\text{text}) \quad (\boldsymbol{X} \in \mathbf{R}^{N \times d_0}) \tag{2}$$

式中 d_0——词向量维度

1.2.2 LSTM 层及其堆叠

由于梯度消失问题，传统 RNN 难以捕捉长距离依赖信息。基于门控结构的 LSTM 能够改善此问题[18-19]。对于每一个时间步长，LSTM 的输出包含两部分：记忆单元和隐藏状态。门控机制用于决定有多少新的输入被写入记忆单元，有多少当前记忆单元的内容被遗忘。

LSTM 凭借门控机制的优点，可以将与记忆单元相关的梯度在很长的时间范围内保持较高值。LSTM 层的隐藏状态输出可以简记为

$$\boldsymbol{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N) = \text{LSTM}(\boldsymbol{X}) \tag{3}$$

式中 $\boldsymbol{H}$——LSTM 隐藏状态输出

$(\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_N)$——$\boldsymbol{H}$ 对应分量

LSTM——LSTM 运算函数

单向的 LSTM 是根据之前的词推测当前的词，但有时候当前词也与之后的词有很大关系，双向 LSTM[20] 可解决此问题，并在需要双向信息的应用中取得了非常大的成效[21]。为了使模型能够学习到序列中前向和后向的信息，本文在模型中引入了双向 LSTM，双向 LSTM 结合了从序列起点开始移动的 LSTM 和从序列终点反向移动的 LSTM。

前向 LSTM 记为

$$\overrightarrow{\boldsymbol{H}} = \overrightarrow{\text{LSTM}}(\boldsymbol{X}) \tag{4}$$

反向 LSTM 记为

$$\overleftarrow{\boldsymbol{H}} = \overleftarrow{\text{LSTM}}(\boldsymbol{X}) \tag{5}$$

最终的输出为前向与后向的拼接

$$\boldsymbol{H} = \overrightarrow{\boldsymbol{H}} \oplus \overleftarrow{\boldsymbol{H}} \tag{6}$$

简记为

$$\boldsymbol{H} = \text{BiLSTM}(\boldsymbol{X}) \tag{7}$$

式中 BiLSTM——双向 LSTM 运算函数

LSTM 通常被描述为具有一定的时间深度，如果将 LSTM 展开，等效于一个隐藏层数与输入序列长度相同的前馈神经网络。这种方式并不涉及太多数据的分层处理，在不同的时间步长上，相同的变换被反复应用于计算输入向量与隐藏层表示[22]。假设输入向量位于相同的表示空间中，则隐藏层输出也位于相同的表示空间中，因此，随着时间步长的增加，并不能得到更加抽象的层次结构表示[23]。为了提高模型的语义表征能力，获得更抽象和更高层次的特征表示，本文在模型中引入了堆叠结构的 LSTM，即把前一层 LSTM 的输出作为输入计算，例如两层堆叠双向 LSTM 可以表示为

$$\boldsymbol{H}_1 = \text{BiLSTM}(\boldsymbol{X}) \tag{8}$$

$$\boldsymbol{H}_2 = \text{BiLSTM}(\boldsymbol{H}_1) \tag{9}$$

式中 $\boldsymbol{H}_1$——第 1 层 LSTM 隐藏状态输出

$\boldsymbol{H}_2$——第 2 层 LSTM 隐藏状态输出

将单层 BiLSTM 转换成两层或者多层结构，通常能带来一定的效果提升[24]，堆叠式的结构也有助于模型学习到更复杂的条件分布，获得更抽象和更高层次的文本表示[25]。对于病虫害文献识别问题，语义特征的表征十分重要。为了进一步加强语义特征的表示能力，除使用堆叠 LSTM 加强抽象表示能力之外，在进行堆叠操作时，通过将输入向量与输出向量拼接，把语义信息附加到单个词的表示中，即

$$\boldsymbol{X} = \text{BiLSTM}(\boldsymbol{X}) \oplus \boldsymbol{X} \tag{10}$$

堆叠层数 n 为上述操作进行的次数。

1.2.3 注意力层

传统的最大池化会丢失特征的位置和强度信息，而平均池化不能很好地突出细节信息。对于病

虫害文献识别问题，摘要文本通常较长，由多个句子组成，不同的词对最终分类结果的贡献不同，具有重要意义的词可能是连续的几个词，也可能分散在文本的不同位置。因此本文采用注意力机制[26]作为池化策略，计算式为

$$\boldsymbol{u}_i = \tanh(\boldsymbol{W}_a x_i + \boldsymbol{b}_a) \tag{11}$$

$$\alpha_i = \mathrm{Softmax}(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_w) = \frac{\exp(\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_w)}{\sum_{j=1}^{N}(\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{u}_w)} \tag{12}$$

式中　$\boldsymbol{W}_a$、$\boldsymbol{b}_a$、$\boldsymbol{u}_w$——可学习的网络参数

α_i——注意力权重　　$\boldsymbol{u}_i$——隐藏层

$\boldsymbol{u}_i$ 可以看成是 $\boldsymbol{x}_i$ 经过一层多层感知机得到的隐藏层表示，然后与一个可学习的上下文向量 $\boldsymbol{u}_w$ 相乘，通过 Softmax 函数求得注意力权重 α_i，$\boldsymbol{u}_w$ 通过随机初始化生成，并在训练过程中更新。最后用加权平均的方式对输入信息汇总，得到嵌入层的输出 $\boldsymbol{A}$ 为

$$\boldsymbol{A} = \mathrm{att}(\boldsymbol{X}) = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i \tag{13}$$

式中　att——注意力运算函数

这实际上是一种“软性”的信息选择机制，每个词的注意力权重可以看成是它对整个文本语义的贡献度。经过注意力层的计算，每个词对于整个文本序列表示的贡献不再完全相同，而是有所侧重，并且所有信息按照其所占权重汇总到一起。基于注意力机制的池化方式使得模型能够决定对哪些文本重点关注，相比于最大池化方法，注意力池化方式可以更多地保留序列中包含的其它信息；相比于平均池化方法，通过将更大的权重分配给更重要的特征，使得模型具有更强的指示性。

1.2.4　Softmax 分类器

得到注意力层的输出之后，在模型顶部添加线性变换层和 Softmax 函数以生成类别空间上的条件概率

$$p = \mathrm{Softmax}(\boldsymbol{W}_o\boldsymbol{A} + \boldsymbol{b}_o) \tag{14}$$

式中　p——条件概率

$\boldsymbol{W}_o$、$\boldsymbol{b}_o$——线性变换层权重参数

然后通过 argmax 函数输出预测标签 $\hat{y}$ 为

$$\hat{y} = \mathrm{argmax}(\boldsymbol{p}) \tag{15}$$

式中　$\boldsymbol{p}$——条件概率向量

损失函数采用二分类交叉熵损失，其计算公式为

$$l_{oss} = -y\ln p_1 - (1-y)\ln(1-p_0) \tag{16}$$

式中　l_{oss}——样本的损失值

y——样本的真实标签

p_0、p_1——向量 $\boldsymbol{p}$ 的第 1 个和第 2 个值

模型的优化目标是使所有训练样本的损失值最小。

1.2.5　正则化与训练

为减少过拟合，在嵌入层之后、每一层 LSTM 层之后以及 Softmax 分类器之前使用了 Dropout，分别设置为 0.2、0.1、0.5。

本文使用 Adam[27]作为优化器，学习率设置为 0.001。模型采用开源深度学习库 TensorFlow（http://www.tensorflow.org/）实现，在 NVIDIA GeForce GTX1660 GPU 上训练。

2　实验结果与分析

2.1　数据集划分

随机选取数据集的 10% 作为测试集，剩余的 90% 中再选取 10% 为验证集，其余为训练集，在选取的过程中保持类别比例一致，划分后各数据集情况如表 1 所示。

表 1　数据集划分情况

Tab.1　Data set division

数据集	正例数	负例数
训练集	1 166	859
验证集	130	95
测试集	143	107

2.2　训练参数设置

由于大部分摘要的序列长度都小于 400，将输入序列长度 N 设置为 400，长度不足 400 的序列以补齐标识符[pad]在其后填充，长度超过 400 的序列截去超过 400 的部分，以使输入序列的长度统一。训练时批尺寸（batch size）设置为 32，训练 25 个周期（epochs）后停止训练，训练过程中在验证集上准确率最高的模型权重参数被保存用于在测试集上进行测试。

2.3　评估指标

对于模型性能的评估，采用标准的评估指标，包括精确率（Precision，P）、召回率（Recall，R）、F1 值和正确率（Accuracy，Acc）作为评价指标。

2.4　LSTM 隐藏层维度选择

LSTM 是模型的基本组件，其性能影响着整个模型的表现，过小的隐藏层维度会使 LSTM 的拟合能力不够，过大的隐藏层维度则会造成过拟合。图 4 为AP－LSTM 模型在不同堆叠层数下，选择不同的隐藏层维度所得到的 F1 值。由图 4 可知，隐藏层维度为 200，能在堆叠层数为 1、2、4 时取得较优效果，因此，本文采用 200 作为 LSTM 隐藏层维度，后续实验中，AP－LSTM 模型的 LSTM 隐藏层维度

均设置为 200。

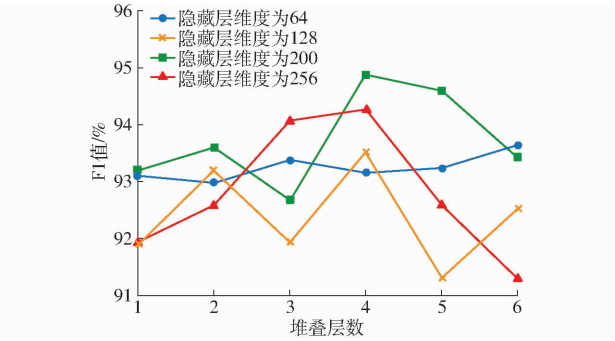

图 4 不同堆叠层数下不同隐藏层维度的 F1 值

Fig. 4 F1 scores of different hidden dimensions under different stacking numbers

2.5 不同池化方式的影响

为了验证基于注意力机制的池化策略对于性能的影响,本文比较了不同池化方式的分类表现。分别使用注意力池化、最大池化、平均池化进行实验,在不同堆叠层数下,不同的池化方式得到的 F1 值如图 5 所示。从图 5 中可以看出,在堆叠层数为 1、2、4、5 时注意力池化方式均取得了最佳表现。最大池化方式在堆叠层数较少时能取得不错的效果,但随着堆叠层数的增加,最大池化方式取得的 F1 值明显低于注意力池化;平均池化方式总体效果比其它两种池化方式差,但在堆叠层数增加时效果有所提升。这是由于更多的堆叠层数使得模型学习到更为抽象的表示,对每个词的表示也更加充分,这种情况下最大池化能获取到的信息比较有限,而平均池化则能更多地利用这些信息,而基于注意力机制的池化方式不但能更多地利用每个词的信息,还能有效地突出重点。从图 4、5 中可以看出,随着堆叠层数的增加,大部分情况下模型能取得更高的 F1 值,但增加到 4 之后会有所下降,且整体最高的 F1 值在堆叠层数为 4 时取得。

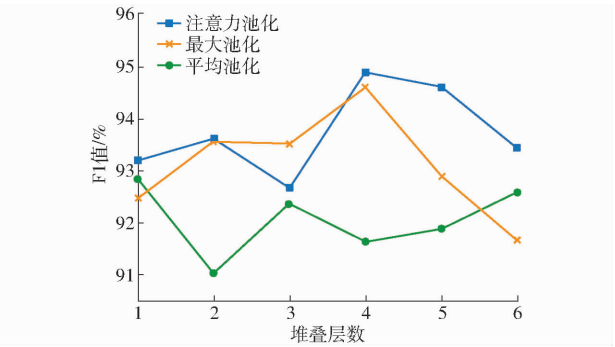

图 5 不同堆叠层数下不同池化方式的 F1 值

Fig. 5 F1 scores of different pooling methods under different stacking numbers

综上所述,后续实验中,AP－LSTM 模型的池化方式均设置为注意力池化,堆叠层数均设置为 4,以使模型达到最佳效果。

2.6 注意力权重可视化

为了说明 AP－LSTM 模型能够捕获摘要文本中具有重要信息的部分,对注意力权重即 α_i 进行了可视化,颜色越深表示 α_i 越大。

图 6 为可视化样例 1,虽然没有出现病虫害之类的指示性词语,但是模型重点关注了对美国白蛾的描述性文本以及概述正文内容的语句“本文重点研究美国白蛾的综合防治技术”,最终预测为正例。

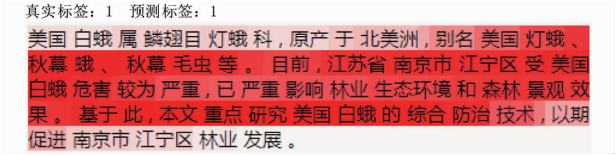

图 6 可视化样例 1

Fig. 6 Visualization example 1

图 7 为可视化样例 2,文本较长,冗余信息较多,AP－LSTM 模型通过注意力机制着重关注了概述正文内容的部分,给出了正确的分类。

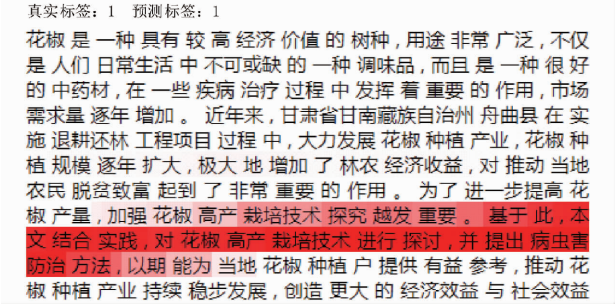

图 7 可视化样例 2

Fig. 7 Visualization example 2

图 8 所示的可视化样例 3 中,“有害生物”作为具有高度指示意义的词得到了重点关注,但是前后文中描述正文内容的一些词组,如“制度”、“动植物检疫法”、“修订”等,也为最终的分类贡献了较大的权重,使得模型没有误判为正例。

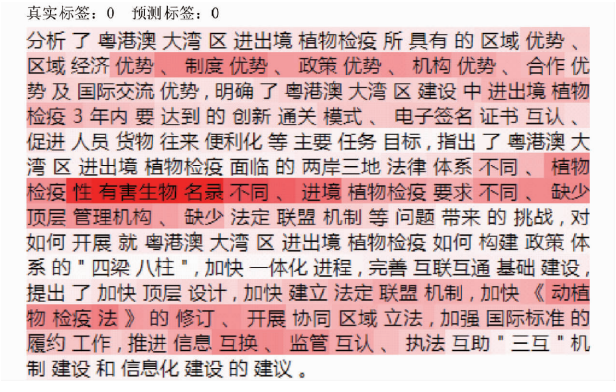

图 8 可视化样例 3

Fig. 8 Visualization example 3

图 9 为另外一个出现了“病害”字样但属于不同领域的例子,模型也给出了正确的分类。

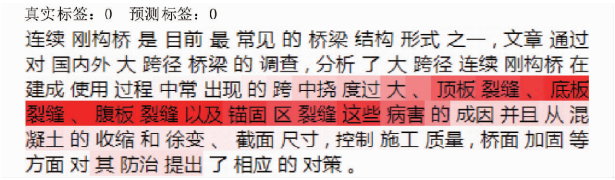

图 9　可视化样例 4

Fig. 9　Visualization example 4

综上所述，AP－LSTM 模型不仅能捕获连续的具有重要信息的文本，而且能捕获分散在不同位置的重要文本，在保留关键词句信息的同时，也能利用其它相关信息，因此可以更好地学习上下文信息，避免冗余信息的影响，缓解指示性词语的误导，从而提高最终的分类效果。

2.7　与基线模型的比较

为验证 AP－LSTM 模型的有效性，选择常用的主流自然语言处理模型作为基线（baseline）模型与 AP－LSTM 模型进行性能对比，包括 TextCNN[28]、BiLSTM、BiLSTM－Attention，实验结果如表 2 所示。

表 2　与基线模型比较的实验结果

Tab. 2　Experimental results compared with baseline model　%

模型	*P*	*R*	F1 值	Acc
TextCNN	86.34	97.20	91.45	89.60
BiLSTM	91.30	88.11	89.68	88.40
BiLSTM－Attention	94.03	88.11	90.97	90.00
AP－LSTM	92.67	97.20	94.88	94.00

由表 2 可知，相比其它基线模型，AP－LSTM 模型在大部分指标上均显著提高。TextCNN 模型也能取得较高的召回率，但精确率较低，这是因为数据集中存在一些带有“病害”、“病虫害”字样的负例样本，TextCNN 模型主要依靠局部结构与目标之间的关联进行分类，极易因为这些指示词的出现而做出正例预测。基于 LSTM 的模型有更强的上下文学习能力，但对于较长的文本，能够学习到的信息有限。结合注意力机制的 BiLSTM－Attention 模型取得了最高的精确率，但召回率较低，因为在一些较长的摘要文本中无法有效避免冗余信息的影响，导致一些正例没有被识别出来。

为进一步说明序列长度的影响，对测试集中不同长度的序列分别进行了实验，测试集中序列长度在 201～300 之间的有 4 条，在 301～400 之间及 400 以上的各 1 条，因此只对序列长度在 200 以下的样本进行了实验。不同模型在不同序列长度样本上的 F1 值如表 3 所示。

表 3　不同模型在不同序列长度样本上的 F1 值

Tab. 3　F1 scores of different models on samples of different sequence lengths　%

模型	序列长度		
	0～50（30 条）	51～100（128 条）	101～200（86 条）
TextCNN	95.00	89.74	91.57
BiLSTM	90.00	92.11	86.75
BiLSTM－Attention	92.68	92.62	87.80
AP－LSTM	95.24	94.27	96.63

由表 3 可知，当序列长度小于 50 时，TextCNN 能取得较好表现，序列长度超过 50 后 F1 值下降明显；当序列长度达到 101～200 时，BiLSTM 和 BiLSTM－Attention 模型的 F1 值明显降低，而 AP－LSTM 模型仍能保持较高的 F1 值。说明 AP－LSTM 模型能够在序列长度较长的情况下进行更有效的识别。

AP－LSTM 模型在不同序列样本上取得了最高的 F1 值、准确率、召回率，精确率也较高，充分说明其在病虫害文献识别上的有效性。

3　结束语

提出了 AP－LSTM 模型，该模型利用堆叠式 LSTM 结构，提高了对语义特征的学习能力，对于每层 BiLSTM，通过将输入向量与输出向量拼接，进一步加强了对语义信息的表征；基于注意力机制的池化策略，使得模型在关注重点词语的同时，也能够按照重要性关注其它有意义的词语，加强了对长上下文信息的利用。在自标注病虫害文献识别数据集上的实验以及可视化效果表明了本文采用的堆叠式 LSTM 结构和基于注意力机制的池化策略有助于提升模型性能，且与其它广泛应用的主流自然语言处理模型相比，AP－LSTM 模型取得了最优的召回率、F1 值和准确率，分别为 97.20%、94.88% 和 94.00%，也取得了较高的精确率，为 92.67%，能够满足实际应用需求。

参　考　文　献

[1]　任妮，孙艺伟，鲍彤，等. 农业领域本体构建方法研究——以番茄病虫害为例[J]. 情报探索，2021(4)：51－57.

[2]　郭旭超，唐詹，刁磊，等. 基于部首嵌入和注意力机制的病虫害命名实体识别[J]. 农业机械学报，2020，51(增刊 2)：335－343.
GUO Xuchao，TANG Zhan，DIAO Lei，et al. Recognition of Chinese agricultural diseases and pests named entity with joint radical embedding and self-attention mechanism[J]. Transactions of the Chinese Society for Agricultural Machinery，2020，51(Supp. 2)：335－343.(in Chinese)

[3] 李想,魏小红,贾璐,等. 基于条件随机场的农作物病虫害及农药命名实体识别[J]. 农业机械学报, 2017,48(增刊):178 - 185.
LI Xiang, WEI Xiaohong, JIA Lu, et al. Recognition of crops, diseases and pesticides named entities in Chinese based on conditional random fields[J]. Transactions of the Chinese Society for Agricultural Machinery, 2017, 48(Supp.):178 - 185. (in Chinese)

[4] 沈利言,姜海燕,胡滨,等. 水稻病虫草害与药剂实体关系联合抽取算法[J]. 南京农业大学学报, 2020, 43(6): 1151 - 1161.
SHEN Liyan, JIANG Haiyan, HU Bin, et al. A study on joint entity recognition and relation extraction for rice diseases pests weeds and drugs[J]. Journal of Nanjing Agricultural University, 2020, 43(6): 1151 - 1161. (in Chinese)

[5] 张林. 跨平台温室生产管理与病虫害决策系统研究与实现[D]. 兰州:兰州理工大学, 2017.
ZHANG Lin. Research and application of cross-platform greenhouse production management and pest decision system[D]. Lanzhou: Lanzhou University of Technology, 2017. (in Chinese)

[6] KIM H, JEONG Y. Sentiment classification using convolutional neural networks[J]. Applied Sciences-basel, 2019, 9(11): 23 - 47.

[7] HE B, GUAN Y, DAI R. Classifying medical relations in clinical text via convolutional neural networks[J]. Artificial Intelligence in Medicine, 2019, 93(SI): 43 - 49.

[8] LIU B, ZHOU Y, SUN W. Character-level text classification via convolutional neural network and gated recurrent unit[J]. International Journal of Machine Learning and Cybernetics, 2020, 11(8): 1939 - 1949.

[9] HU H, LIAO M, ZHANG C, et al. Text classification based recurrent neural network[C]// 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020.

[10] ABDI A, SHAMSUDDIN S M, HASAN S, et al. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion[J]. Information Processing & Management, 2019, 56(4): 1245 - 1259.

[11] YU Y, SI X, HU C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural Computation, 2019, 31(7): 1235 - 1270.

[12] DU J, GUI L, XU R, et al. A convolutional attention model for text classification[C]// Natural Language Processing and Chinese Computing, NLPCC 2017: Springer International Publishing Ag, 2017: 183 - 195.

[13] 卢玲,杨武,王远伦,等. 结合注意力机制的长文本分类方法[J]. 计算机应用, 2018, 38(5): 1272 - 1277.

[14] 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, 2020, 57(15): 1 - 8.

[15] WANG Z. Attention-based bilstm fused CNN with gating mechanism model for Chinese long text classification[J]. Computer Speech & Language, 2021(6): 101182.

[16] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]// Advances in Neural Information Processing Systems NIPS, 2013.

[17] LAI S, LIU K, HE S, et al. How to generate a good word embedding[J]. IEEE Intelligent Systems, 2016, 31(6): 5 - 14.

[18] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735 - 1780.

[19] GERS F, FELIX A, SCHMIDHUBER J, et al. Learning to forget: continual prediction with LSTM[J]. Neural Computation, 2000, 12(10): 2451 - 2471.

[20] SCHUSTER M, PALIWAL K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673 - 2681.

[21] GRAVES A. Supervised sequence labelling with recurrent neural networks[J]. Studies in Computational Intelligence, 2012, 385(6): 37 - 45.

[22] IRSOY O, CARDIE C. Opinion mining with deep recurrent neural networks[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[23] BENGIO Y. Learning deep architectures for AI[M]. Now Publishers Inc., 2009.

[24] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]// Advances in Neural Information Processing Systems NIPS, 2014.

[25] KONG X, TU Z, SHI S, et al. Neural machine translation with adequacy-oriented learning[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2019: 6618 - 6625.

[26] DZMITRY B, KYUNGHYUN C, YOSHUA B. Neural machine translation by jointly learning to align and translate[EB/OL]. [2016 - 05 - 09]. https://arxiv.org/abs/1409.0473v2.

[27] KINGMA D, JIMMY B. Adam: a method for stochastic optimization[C]// International Conference on Learning Representations (ICLR), 2015.

[28] KIM Y. Convolutional neural networks for sentence classification[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.