基于 TFIDF 与 XGBoost 的新闻分类研究

彭博 吕文英 熊睿

南宁学院人工智能学院，南宁，530200　　北京资采信息技术有限公司，北京，100071

摘　要　文本分类是自然语言处理技术在工程当中的重要应用。随着现代机器学习研究的不断发展和提升,自然语言处理技术能够结合各种机器学习算法，并以其较强的可解释性及在文本分类任务当中获得较好的分类效果而广泛应用在各种文本分类场景。本文主要研究针对新闻网站提供出来的新闻文本数据，使用 TFIDF 方法来提取文本特征信息，根据所提取到的包含特征信息的特征向量使用一些经典机器学习算法如逻辑回归算法(LR)、支持向量机算法(SVM)以及集成学习算法随机森林算法(RF)、XGBoost 构建分类模型对各算法的分类效果进行分比对分析。结果表明，TFIDF 与 XGBoost 进行新闻分类的效果对比其他三种分类算法平均值准确率高出 9.16%，分类效果较好。

关键字　自然语言处理，机器学习，文本分类，文本向量化

Research on News Classification Based on TFIDF and XGBoost

Peng Bo　Xiong Rui　　Lv Wenying

School of Artificial Intelligence
Nanning University
Nanning 530200, China
bopeng8@iflytek.com；ruixiong2@iflytek.com

Beijing Zicai Information Technology Co. LTD
Beijing 100071, China
569153677@qq.com

Abstract—Text classification is an important application in project of natural language processing technology. With the development of modern machine learning methods, NLP processing and analysis techniques will be combined with various machine learning algorithms because of its strong interpretability and good classification results in text classification tasks. This paper mainly focuses on the news text data provided by news websites, and uses TFIDF method to extract text feature information. According to the extracted feature vectors containing feature information, some classical machine learning algorithms such as Logistic Regression (LR), Support Vector Machine (SVM), ensemble learning algorithms Random Forest (RF) and XGBoost were used to construct classification models to compare and analyze the classification effects of each algorithm. The results show that the average accuracy of TFIDF and XGBoost for news classification is 9.16% higher than that of the other three classification algorithms, and the classification effect is good.

Key words—Natural Language Processing, Machine Learning, text classification, text vectorization

1　引　言

在网络数据资源中，数据大致可以分为共三种，分别为文本数据、音频数据和图片视频数据[1]，其中占比最多的形式是文本数据，现今，新闻网站的后台需要耗费大量人力管理海量新闻数据，不仅人力成本随之提高，大众按照个人需求能获取到有用新闻的难度也增加了[2]。在新闻数据量与用户数的与日俱增前提下，可以将文本分类技术应用于互联网的新闻数据分类处理，以解决海量数据造成的上述问题。

文本分类技术是从属于自然语言处理（Natural Language Processing, NLP）领域之中，应用较热门的研究技术，国内外也有较多 NLP 行业内的学者、专家对文本分类有着较深入的探索和研究。国外对于自然语言处理领域发力较早。早期采用“专家系统”，使用制定规则的方法，研究组合类别关键字来达成对文本进行分类的诉求。这种方法虽然在可行性方面有一定的操作空间，但其缺陷也是相当的明显，该方式通用性较差，不同任务之间无法复用专家制定的分类规则。因此专家往往需要根据不同的任务动态调整不同的规则，大量的时间消耗在更改、甚至重制规则上面。

之后，随着计算机技术深耕于硬件的研发，在较高水平硬件支撑下，机器学习理论被应用于数字图像识别和图像分类领域，在图像处理中的可用性和有效性带给了人们使用经典机器学习算法进行文本处理的信心。随后机器学习也逐渐被应用在处理文本数据的场景。应用机器学习的文本分类技术核心为两个模块，一个模块是特征提取，另一个模块是分类器[3]。特征提取的目的及作用，是将文本数据转换为向量的方式来进行表示和处理，也称文本向量化，这种文本向量化的方式，能最大限度地反映文本的数据含义，其文本含义以数学形式进行了转化、保留。特征提取步骤

十分重要，特征提取的质量将对文本分类的效果起到十分关键的作用，可以决定最终结果的上限。分类器的作用是根据提取的文本特征，给出目标文本的类别信息，机器学习算法处理文本分类是在特征提取的基础上，引入诸如逻辑回归分类器和支持向量机分类器，把文本特征向量输入分类器、分类结果作为输出进行训练[4]。机器学习的文本分类技术也取得了比较好的分类效果，逐渐被企事业单位、政府机构等作为常规工具来进行使用来减小人工进行分类成本。

除机器学习外，深度学习的高速发展给 NLP 领域注入了新动力，带来了新的可能。对于文本分类任务，深度学习与机器学习相比在文本特征表示及分类器选择均有不同。深度学习采用 Word Embedding[5]的方式把文本表示为词向量方式进行特征提取，相较于经典的特征提取方式 TFIDF，Word Embedding 有着维度低的特点。深度学习常用的文本分类器包括卷积神经网络（CNN）[6]、循环神经网络（RNN）[7]、双向长短时记忆网络（BiLSTM）等。CNN 擅长于捕获局部的文本信息，RNN 可以顺序接收文本中的单词，网络在时间轴上表现出具有一定的记忆性。而 BiLSTM 是在 RNN 的基础上引入多种门单元，能够解决序列中的长期依赖问题。现阶段大热的自然语言处理 Bert 模型[8]，加入了 Attention 注意力机制，采用了 Transformer 作为特征提取器，在 NLP 领域有着非常广泛的应用和优秀效果。而国内的文本分类研究相对起步较晚，但在一众优秀的研究人员的努力下，国外研究基础上进行了模型融合[9]、优化[10]，取得了不错的成绩和效果。

诚然，现在深度学习在文本分类中的表现十分亮眼，但其网络复杂性、对计算资源的大量消耗以及神经网络独有的可解释性较弱的特点仍在会在一些文本分类场景下仍然客观存在。而在机器学习领域，近几年机器学习的新星，XGBoost 作为当前机器学习算法的“大杀器”，在众多算法竞赛中取得了骄人的战绩，在分类效果上也可以深度学习模型一较高下，但其应用在文本分类的研究却是相对较少。基于此，本文选择了使用 TFIDF 进行文本特征提取，结合 XGBoost 来实现新闻文本的分类，相对于神经网络方法，提高了算法的可解释性、节省了计算资源，相对于一些经典机器学习算法，则提高了新闻分类的准确率。

2 基于 TFIDF 与 XGBoost 的新闻分类方法

2.1 文本预处理

进行中文文本分类的第一步，就是文本预处理。分词、去停用词等过程，是文本预处理的一般流程。若文本中含有大量无意义字符、数字，在文本预处理中也应将其去除。总的来说，尽可能去除文中的噪声，提高分类的准确率是文本预处理的主要目的及意义。

（1）中文分词

中文文本与英文文本的分词虽同为分词操作，但其差异相对较大。空格作为英文单词之间的天然分隔符，使得英文的分词可以通过空格实现对各单词的划分与识别；中文文本各词语间则没有英文这种表示，无明显的分隔符号，为对中文文本中用到的词汇进行统计，则先要对文本进行分词处理。因同一词汇在句子中的位置不同，可能表示不同含义，故分词的准确对文本分类性能影响较大。如今普遍使用的中文分词方法主要有以下三种：基于字符匹配的方法、基于统计的方法以及基于理解的方法。各分词法各有其优点亦有其弊端，实际应用中可采用将几种方法结合的方式进行分词。

（2）去停用词

停用词指在处理文本前预先过滤掉的某些字、词或特殊字符。停用词大体分为两类：一类是高频或使用广泛的词，因为使用频次过高，无法根据该词对分类结果给出有效判定；另一类是无明确意义的词，包括副词、连接词、标点符号等。为提高分类准确率，需要将这两类停用词去除。在实际应用中，我们通常使用一个停用词表，将分词结果中与停用词表中相同的字或词进行滤除操作。

2.2 特征提取

文本预处理好得到分词结果后，即可执行后续特征提取的操作。TFIDF 也被称作词频与逆文档频率，是一种提取文本特征的统计方法。实际上 TFIDF 是词袋模型的一种，TF 是 Term Frequency 两个单词的首字母缩写，表示词频；IDF 则是 Inverse Document Frequency 的首字母缩写，表示逆文档频率。TFIDF 的计算公式如下：

$$TFIDF_{ij} = \log\left(f_{ij}\right) \times \log\left(\frac{N}{n_j}\right) \qquad (1)$$

公式（1）中，f_{ij} 即表示的是词的频率即词频，代表词 j 在文档 i 中总体的使用频率，即词 j 在文档 i 中出现次数与文档 i 中总共所有的词数的比。N 代表文档的总数即语料中共包含多少个文档，n_j 代表出现词 j 的文档个数。上述公式表示的含义，是当一个词在一篇文章中出现的频次多，并且该词语在其他文章中出现的次数少时，$TFIDF_{ij}$ 的值是比较大的，也就是表示这个词 j 对这篇文章 i 比较重要。解决的是当一个文档中有相同词频的不同词时，区分这些词对文档的重要性问题。例如，词 a 与词 b 在文档 i 中词频相同，此

时仅用词频来计算词的重要性是不能区分两词的重要性的。通过逆文档频率的引入来计算这两个词在整个文档集中的逆文档频率，若词 a 在所有文档集出现的次数较少，而词 b 在所有文档集中出现的次数较多，那么就说明，词 a 拥有比词 b 有更好的文档区分能力，这就是逆文档频率的作用。对文档 i 来说，词 a 的 TFIDF 的值就会比词 b 的 TFIDF 的值大。

由此可见 TFIDF 用来评估一个字词对于文件的集合或语料库中的一份文档或一个类别的重要程度。其主要思想是：如果某个词或短语在一个类别中出现的频率较高，并且在其他类别中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类[11]。

2.3 XGBoost

XGBoost（极端随机树）[12]是由陈天奇博士等人共同提出研究的一种以集成算法思想理论为技术基础的一种机器学习算法，通过多个弱学习器比单个弱学习器具有更高的精度的特点来进行建模。与传统的集成学习方法不同，如随机森林算法是采用装袋法（bagging）思想，XGBoost是采用一种并行方式产生子树且最后通过投票的方法集成产生的所有结果。而XGBoost采用提升法boosting思想，通过串行方式产生子树最终通过加权的方式集成所有结果。而且XGBoost与同样采用boosting思想的梯度提升树GBDT算法不同之处在于，梯度提升树算法在优化时即在与真实值逼近的过程中使用的是一阶导数信息，而XGBoost使用二阶泰勒展开使其逼近过程中更加精确、快速。作为机器学习策略中的另一种,XGBoost通过将多颗决策树中所有的预测树结果加权计算从而可得出最终结果的预测结果,公式如下：

$$y_j = \sum_{l}^{L} f_l(x_j), f_l \in F \qquad (2)$$

公式（2）中，y_j即为该模型预测最终的结果，L 表示单一的弱学习器即单一决策树的数量（n_estimators），f_l为第l颗树，x_j表示第j个输入样本，F表示所有树模型的集合。XGBoost在上述理论基础上还专门研究利用了贪心算法来遍历决策树模型下的未实现分裂的叶子节点,目标是为了获取对分裂后的目标函数增益系数为最大值的那一个叶子节点来进行分裂。XGBoost中，树的个数、深度、叶子节点最小样本数等参数均会对预测结果产生影响。

2.4 本文的方法

本文的主要思路及流程，即按照文本分类的一般处理流程，首先进行文本数据预处理；然后使用TFIDF提取预处理后文本特征，将文本数据转换为特征矩阵；最后将包含文本特征的特征矩阵送入XGBoost模型中，训练并得到分类模型。

（1）文本数据预处理

本文进行预处理的分词、去停用词的操作是采用python中的jieba分词工具来实现的。jieba分词结合了两种常用的分词方式----基于字符串匹配的算法和基于统计的算法，使用了最大概率路径动态规划算法来匹配字符串，提高分词速度与精度；同时使用隐马尔科夫（HMM）模型对未登录词（新词）进行分词，有效的解决了字符串匹配无法识别未登录词（新词）的痛点。

（2）使用 TFIDF 进行特征提取

对于预处理好后的文本，即可使用 TFIDF 对其进行特征提取。

TFIDF 算法，实际上可以看成有 TF 与 IDF 组合而成。TFIDF 在提取词的特征时，主要是根据词在文档中出现的次数即词频 TF 和逆文档频率 IDF 来计算词重要性。若一个句子中某词出现 3 次，可认为这个词在该句中是一个显著特征，但是该词在其他句子中出现了 8 次，这样的话显然这个词能否作为重要特征，就要结合整篇文档考虑来看了。

TF：这个是用于表示一个词在文本中出现的频率数，一般某词的 TF 越大，句中该词出现次数越多，我们认为该词重要性越高。

IDF：这个是改变词的权重的的一个频率。 对于某些虚词如“的”、“地”和“得”，在句中出现的频率很高，但没有实际的语义，这些噪声会影响我们对文本特征重要性的一个判断，因此引入 IDF，为这些单词进行重要性降权。

由此，通过 TFIDF 算法，我们就可以计算出句子中单词的重要性程度，而该算法在本实验中，我们可以直接使用利用 sklearn 中的 TfidfVectorizer 类来实现 TFIDF 特征提取，得到的，就是文本的特征矩阵，具体特征提取代码如下：

```
# 训练集数据的读取
train_data = pd.read_csv('../data/sohu_train.txt',
                                    sep='\t',
                                    header=None,
                                    dtype=np.str_,
                                    encoding='utf8',
                                    names=[u'频道', u'文章'])


# 载入停用词
stopwords = set()
with open('../data/stopwords.txt', 'r', encoding='utf-8') as infile:
    for line in infile:
        line = line.rstrip('\n')
        if line:
            stopwords.add(line.lower())
```

```
# 实例化特征提取对象，提取文本特征
tfidf = TfidfVectorizer(tokenizer=jieba.lcut,
                        stop_words=stopwords,
                        min_df=50,
                        max_df=0.3)
x = tfidf.fit_transform(train_data[u'文章'])
# 查看特征提取后的特征矩阵
print(x.toarray())
# 输出TFIDF模型词汇表大小
print('词汇表大小为：{}'.format(len(tfidf.vocabulary_)))
```

（4）XGBoost 算法设计

使用 TFIDF 提取了新闻文本特征后，接下来就可以拆分训练集、测试集，然后送入 XGBoost 模型中进行建模、验证了。XGBoost 算法实质是基于 boosting 和 CART 决策树的集成算法，用作分类场景，就是构建 n 颗 CART 树，并尽量减少树群的预测误差，而且要尽可能拥有较强的泛化能力与鲁棒性。此处我们可以使用 python 的 xgboost 来实现极端提升树算法，这里我们主要设置一些算法参数，来具体设计、实现 XGBoost。下面对一些关键参数的功能及赋值，做一下说明：

Booster：gbtree 基分类器使用树模型，为默认参数值；gbliner 表示基分类器为线性模型。

n_estimatores：基分类器数，即决策树的个数。

max_depth：此参数默认初始值为 6。这个值决定了单颗树的最大深度，且值越大，越容易过拟合。

learning_rate：学习率，每次迭代更新可使用该参数控制更新权重时的步长，默认 0.3。该参数特点是值越小，训练越慢，学习率的典型值一般为 0.01-0.2。

objective ：指定目标函数，需根据具体任务使回归、二分类、多分类设置不同的参数值。

XGBoost 关键参数取值可以采用交叉验证、网格搜索等方式不断调参，确定效果较好的参数取值。本实验中取得效果较好的参数取值如表 1 所示：

表 1 XGBoost 训练参数

参数	初始值
objective	multi:softmax num_class=12
n_estimators	100
max_depth	4
learning_rate	0.1
min_child_weight	1
booster	gbtree

使用表中参数设计XGBoost分类算法，即可构造分类模型，并查看新闻分类效果，核心代码如下：

```
# 采用分层抽样方式来划分训练集与测试集，抽样更为科学
train_idx, test_idx = train_test_split(range(len(y)),
                                       stratify=y,
test_size=0.2)
# numpy索引、切片获取目标数据
train_x = x[train_idx, :]
train_y = y[train_idx]
test_x = x[test_idx, :]
test_y = y[test_idx]

import xgboost as xgb
model = xgb.XGBClassifier(max_depth=4,
                          learning_rate=0.1,
                          n_estimators=100,
                          objective='multi:softmax
num_class=12')
model.fit(train_x, train_y)

# 应用模型预测结果到测试集当中
test_y_pred = model.predict(test_x)


# 混淆矩阵的计算,目的为了查看分类效果
cm = pd.DataFrame(confusion_matrix(test_y, test_y_pred),
                  columns=y_encoder.classes_,
                  index=y_encoder.classes_)
```

3 实验过程及结果

本次实验所采用的开发环境为 Pycharm，python 版本为 3.6.8。实验数据使用搜狐新闻网的新闻数据，获取新闻文本及其对应的类别信息，共计 36000 条新闻文本及 12 类新闻分类，包括体育、财经、健康、娱乐、教育、文化等，将其中 24000 条新闻及类别标签作为训练集训练分类模型，12000 条新闻文本及标签作为验证集验证、比对不同算法之间的分类效果。

（1）数据初探、预处理

在使用机器学习进行文本分类工作之前，也需要对获取的文本数据进行数据初探操作，若发现数据质量不高，则需对数据进行清洗，清洗后才能进行后续如特征工程等操作。本实验采用了 python 的 jieba 库，对获取的新闻文本进行分词操作。对于去停用词，本实验使用的停用词表包含有 1452 个词，包括一些特殊符号、低频词语以及无意义的词等，除此之外，也可根据不同语料库和个人实验的需要，增加个性化停用词，适配当前项目，建立适合实验需要的停用词表。根据停用词表的不同类别做出部分停用词，表 2 所示：

表 2 部分停用词表

低频词	特殊符号	无意义词
罢了	$	的
哉	%	吧
譬喻	*	和
设若	#	呢

（2）特征提取

文本数据属于非结构化数据，一般要转换成结构化的数据，将文本数据特征进行向量化[13]。在完成了对新闻文本数据的预处理分析后,接下来任务就是要将分词处理后生成的新闻文本数据进行文本向量化即特征提取，提取文本特征后送入机器学习模型中训练分类模型。本实验对于新闻的文本数据，采用的是 TFIDF 的方式提取特征，利用 sklearn 中的 TfidfVectorizer 类的实例化对象来实现文本特征的提取，通过设置 max_df、min_df 参数过滤常用词、生僻词，得到更好的特征提取效果。

（3）模型评价指标

对于分类任务，常用评价指标有如下几个：

精确率：表示预测为正的样本中预测正确的比率

$$P = \frac{T_p}{T_P + F_P} \qquad (3)$$

召回率：原为正的样本中预测正确数目的比率

$$R_{TPR} = \frac{T_P}{T_P + F_N} \qquad (4)$$

F1 分数：表征精确率与召回率的调和平均数

$$F_1 = 2\frac{PR_{TPR}}{P + R_{TPR}} \qquad (5)$$

准确率，即预测正确的样本占样本总数的比例

$$R_{ACC} = \frac{T_P + T_N}{T_P + F_P + F_N + T_N} \qquad (6)$$

（4）模型构建及结果

完成文本向量化，提取新闻文本特征后，选择经典机器学习分类算法支持向量机、逻辑回归、随机森林算法及 XGBoost 算法分别进行模型构建，几种机器学习算法分类准确率如表 3 所示：

表 3 分类算法结果比对表

算法名称	准确率
支持向量机	77.78%
逻辑回归	85.84%
随机森林	88.96%
XGBoost	93.36%

在实验结果中可以看出，XGBoost（极端随机树）结合 TFIDF 效果最好，在当前新闻文本分类案例中有最优的分类效果。根据该结果，我们使用模型评估报告来查看 XGBoost 结合 TFIDF 在不同新闻类别的分类效果如何，查看算法结果核心代码如下：

```
# 分类效果
def eval_model(y_true, y_pred, labels):
    # 计算每个分类的 Precision、Recall、f1、support
    p, r, f1, s = precision_recall_fscore_support(y_true, y_pred)
    # 计算总体的平均 Precision、Recall、f1、support
    tot_p = np.average(p, weights=s)
    tot_r = np.average(r, weights=s)
    tot_f1 = np.average(f1, weights=s)
    tot_s = np.sum(s)
    res1 = pd.DataFrame({
        u'Label': labels,
        u'Precision':p,
        u'Recall':r,
        u'F1':f1,
        u'Support':s
    })
    res2 = pd.DataFrame({
        u'Label':[u'总体'],
        u'Precision': [tot_p],
        u'Recall': [tot_r],
        u'F1': [tot_f1],
        u'Support': [tot_s]
    })
    res2.index = [999]
    res = pd.concat([res1, res2])
    return res[[u'Label', u'Precision', u'Recall',u'F1',u'Support']]
eval_result = eval_model(test_y, test_y_pred, y_encoder.classes_)
print(eval_result)
```

将分类结果整理后，具体结果见表 4：

表 4 TFIDF+XGBoost 模型新闻分类结果

Label	Precision	Recall	F1
体育	0.9801	0.9875	0.9838
健康	0.9649	0.9625	0.9637
女人	0.9184	0.9575	0.9375
娱乐	0.9131	0.9725	0.9418
房地产	0.9213	0.9375	0.9293
教育	0.9696	0.9575	0.9635
文化	0.9119	0.8800	0.8956
新闻	0.8693	0.8650	0.8671
旅游	0.9579	0.9675	0.9626
汽车	0.9676	0.9725	0.9701
科技	0.9226	0.8950	0.9086
财经	0.9066	0.8500	0.8774
总体	0.9336	0.9337	0.9334

综上，TFIDF+XGBoost 的分类准确率，比使用集成学习的随机森林算法，要高出 4.40%，对比其他三种分类算法平均值准确率高出 9.16%。由上表也可知在大多数类别的新闻中 TFIDF+XGBoost 都能够有非常好的分类效果。

4 结束语

本文在现有的各种文本分类特征提取技术及文本分类算法中，选择了 TFIDF 结合 XGBboost 的方法来完成新闻文本分类任务，与深度学习相比，该种方法无需依赖 GPU 的计算资源支撑，也提升了算法整体的可解释性；对比经典的机器学习算法，提升了分类的效果，也给出了机器学习算法中应用在文本分类场景较少的 XGBoost 算法进行新闻文本分类的一种思路。

参 考 文 献

[1] Alexanderson S, Székely É, Henter G E, et al. Generating coherent spontaneous speech and gesture from text[C]// Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents. 2020: 1-3.

[2] Spinde T, Rudnitckaia L, Mitrovi J, et al. Automated identification of bias inducing words in news articles using linguistic and context-oriented features[J]. Information Processing & Management, 2021, 58(3): 102505.

[3] Deepa D, Raaji, Tamilarasi A. Sentiment Analysis using Feature Extraction and Dictionary-Based Approaches[C]// 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE, 2019: 786-790.

[4] 王润民，桑农，丁丁，陈杰，叶齐祥，高常鑫，刘丽. 自然场景图像中的文本检测综述[J]. 自动化学报，2018, 44(12): 2113-2141.

[5] Gupta S, Kanchinadam T, Conathan D, et al. Task-optimized word embeddings for text classification representations[J]. Frontiers in Applied Mathematics and Statistics, 2020, 5:67.

[6] Wang H, He J, Zhang X, et al. A Short Text Classification Method Based on N-Gram and CNN[J]. Chinese Journal of Electronics, 2020, 29(2):248-254.

[7] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network[J]. Physica D: Nonlinear Phenomena, 2020, 404: 132306.

[8] Yang K, Xu H, Gao K. CM-BERT: Cross-Modal BERT for Text-Audio Sentiment Analysis[C]// Proceedings of the 28th ACM International Conference on Multimedia.2020: 521-528.

[9] 吴汉瑜，严江，黄少滨，等. 用于文本分类的CNN_BiLSTM_Attention 混合模型[J]. 计算机科学2020,47(z2).

[10] 张小为,邵剑飞.基于改进的BERT-CNN模型的新闻文本分类研究[J].电视技术,2021,45(7).

[11] 丁月，汪学明. 一种基于改进特征加权的朴素贝叶斯分类算法[J/OL].计算机应用研究,2019,36(12).

[12] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. CoRR, 2016.

[13] 孟涛，王诚.基于扩展短文本词特征向量的分类研究[J/OL]. 计算机技术与发展，2019(04).