多类别文本分类方法比较研究

于卫红
(大连海事大学 航运经济与管理学院,辽宁 大连 116026)

摘 要:文本分类特别是多类别文本分类问题是非常重要的经典问题,在舆情监测、新闻推荐、在线评论情感分析等领域有着广泛的应用。目前,可用于多类别文本分类的算法很多,但每个算法都有其特定的假设和优缺点。为了帮助使用者或研究者更好地选择和改进分类方法,设计了多类别文本分类方法比较方案,综合考虑了文本特征表示方法和分类算法两个维度,对 3 种文本特征表示方法和 5 种分类算法进行组合,形成 15 种分类模型作为比较对象。基于所设计的比较流程,以从媒体阅读网站 SKIP-GRAM 爬取 SKIP-GRAM 的 3 000 条不同类别的资讯文本为研究语料,对 15 种模型在不同数据规模下进行若干次比较后,以 Kappa 系数和运行时间作为评估指标。综合评估后认为:使用词嵌入进行文本特征表示无论在分类模型的运行速度上还是分类效果上都具有明显的优势,KNN+CBOW、SVM+CBOW、朴素贝叶斯+CBOW 都是解决多类别文本分类问题较佳的模型。

关键词:文本分类;多类别;机器学习;文本特征表示;分类算法
中图分类号:TP391.1 **文献标识码**:A **文章编号**:1673-629X(2022)01-0054-07
doi:10.3969/j.issn.1673-629X.2022.01.010

Study on Comparison of Multi-class Text Classification Methods

YU Wei-hong
(School of Transportation Economics and Management, Dalian Maritime University, Dalian 116026, China)

Abstract: Text classification, especially multi-class text classification, is a classical problem of great significance, which has a wide range of applications in the fields of public opinion monitoring, news recommendation, online comment sentiment analysis and so on. At present, there are many algorithms for multi-class text classification, but each algorithm has its own specific assumptions and advantages and disadvantages. To help users and researchers better choose and improve the classification methods, a comparison scheme based on multi-class text classification is designed. Considering text feature representation and classification algorithm, three text feature representation methods and five classification algorithms are combined to form 15 classification models which are ranked by the comparison scheme. Using 3 000 documents with different categories crawled from media sites as the corpus, these 15 combinations are compared with different scale of data following the process established and are ranked by Kappa coefficient and running time. It is concluded that word embedding for text feature representation has obvious advantages both in the running speed of the model and the classification performance. Meanwhile, KNN+CBOW, SVM+CBOW and Naive Bayes+CBOW are all better models for solving multi-class text classification problems.

Key words: text classification; multi-class; machine learning; text representation; classification algorithm

0 引 言

文本分类是指对于一个特定的文档,判断其是否属于某个类别[1]。根据目标类别的不同,通常将文本分类问题分为三种类型:

(1)二分类:表示分类任务中有两个类别(0 或者 1),如垃圾邮件分类。

(2)多类别分类:表示分类任务中有多个类别,如客户的评论情感可分为 5 个类别:非常满意、满意、一般、不满意、非常不满意。

(3)多标签分类:表示给每个样本分配一个标签集。如,一个文本可能被同时认为是与宗教、政治或教育都相关的话题,或全部无关。

在文本挖掘的实际应用中,多类别分类问题更加常见,并且,多标签分类问题也可以转化为多类别分类问题来加以解决。多类别分类问题较之二分类问题更加复杂,如何选择合适的算法,构建出性能较优的多类别分类模型至关重要。

决策树、随机森林、朴素贝叶斯等算法都可用于多

收稿日期:2021-02-07 **修回日期**:2021-06-09
基金项目:辽宁省社科规划基金项目(L17BGL025)
作者简介:于卫红(1972-),女,博士,副教授,研究方向为多 Agent 理论与应用、智能信息处理、文本挖掘。

类别分类问题,但每个算法都是基于某些特定的假设的,都具有各自的优缺点,没有任何一种分类算法可以在所有的问题解决中都有良好的表现。因此,只有比较了多种算法的性能才能为具体的问题选择出较佳的模型。

1　文本分类的流程、方法与性能评价指标

1.1　文本分类的流程

如图 1 所示,无论何种类型的文本分类问题,其处理过程大都包括文本预处理、文本特征表示、分类模型构建、模型评估几个步骤。其中,文本特征表示和分类模型的构建是文本分类问题的核心。

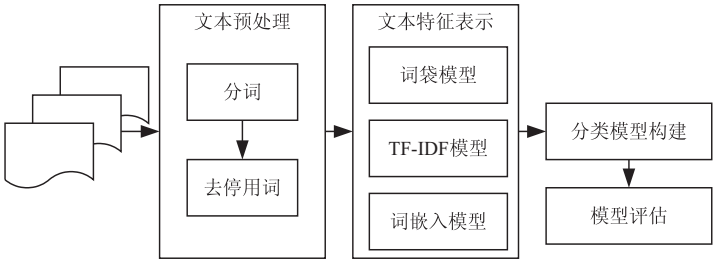

图 1　文本分类的流程

1.2　文本特征表示的几种方法

1.2.1　词袋模型

词袋模型是一种基于词频的对文档进行特征提取的方法,即将文档看作词的集合,对文档中出现的所有词进行词频统计,用词频向量来表示文档[2]。词袋模型忽略了文本的语法和语序等要素,只考虑词在文档中出现的次数。

1.2.2　TF-IDF 模型

TF-IDF 模型在考虑词频的基础上考虑了词对于一篇文章的重要性。TF(term frequency)指的是一个单词在某个文档中出现的频率。通常,一个词在一篇文档中出现的频率越高,这个词对于该文档越重要。IDF(inverse document frequency)指的是逆向文档频率,代表了词对于文档的区分度,如果一个词在一篇文档中多次出现,但在其他文档中很少出现,则认为这个词对于该文档的区分能力较强[3]。一个词的 TF-IDF 值的计算公式为:

$$\text{TF-IDF}=\text{TF}*\text{IDF};$$

$$\text{TF}=\frac{\text{词在文档中出现的次数}}{\text{文档的总次数}};$$

$$\text{IDF}=\log\frac{\text{语料库中文档总数}}{\text{包含该词的文档数}+1} \tag{1}$$

1.2.3　词嵌入模型

基于词嵌入的文本特征表示是一种文本深度表示模型,其主要思想是将文本转换为较低维度空间的矢量表示[4]。首先基于大量的语料库训练出词嵌入模型,即将每个词映射成 K 维实数向量(通常 $K=50\sim200$)[5],并且使得这些向量能较好地表达不同词之间的相似和类比关系,以引入一定的语义信息。常用的词嵌入算法有 Word2Vec 和 Glove[6]。本研究使用 Word2Vec 算法,Word2Vec 有两种实现词嵌入的方式,即 CBOW(连续词袋)和 SKIP-GRAM(跳字模型)[7]。CBOW 方法以上下文单词作为输入,预测目标单词;而 SKIP-GRAM 方法以目标单词作为输入,预测单词周围的上下文。最后,基于训练好的词嵌入模型,使用 Doc2Vec 算法生成文本的向量表示模型,即将每个文本映射成 K 维实数向量。

1.3　构建文本分类模型的常用算法

构建文本分类模型的算法有很多,如传统算法:决策树、多层感知器、朴素贝叶斯、逻辑回归和 SVM;集成学习算法:随机森林、AdaBoost、lightGBM 和 xgBoost;以及深度学习算法:前馈神经网络和 LSTM。对所有算法进行比较,工作量巨大,本研究只比较常用的 5 种算法:决策树、KNN、朴素贝叶斯、SVM 和随机森林。

1.3.1　决策树

决策树是一种以树形结构来展示决策规则和分类结果的模型[8],其思想是通过 ID3、C4.5、CART 等算法将看似无序、杂乱的训练数据转化成可以预测未知实例的树状模型[9]。决策树中每一条从根节点(对最终分类结果贡献最大的属性)到叶子节点(最终分类结果)的路径都代表一条决策规则。

1.3.2　KNN

KNN 算法又称 K 邻近算法、K 最近邻算法,其核心思想是如果一个样本在特征空间中的 K 个最相邻的样本中的大多数属于某一个类别,则该样本也属于这个类别,并具有这个类别上样本的特性[10]。

1.3.3　朴素贝叶斯

朴素贝叶斯算法的核心思想非常朴素:对于给出的待分类项,求解在此项出现的条件下各个类别出现的概率,哪个最大,就认为此待分类项属于哪个类别[11]。

1.3.4　SVM

SVM 即支持向量机算法,最初提出是为了解决二

分类问题,核心思想是基于训练集在样本空间中找到最优的一条线(超平面),将不同类别的样本分开[12]。所谓的“支持向量”就是那些落在分离超平面边缘的数据点形成的线。SVM 算法也可以用于解决多类别分类问题[13],此时,支持向量机仍将问题视为二分类问题,但会引入多个支持向量机用来两两区分每一个类,直到所有的类之间都有区别。

1.3.5 随机森林

随机森林是一种集成学习算法,通过构建并结合多个学习器来完成学习任务[14]。随机森林的出现主要是为了解决单一决策树可能出现的很大误差和过拟合的问题[15],其核心思想是将多个不同的决策树进行组合,利用这种组合降低单一决策树有可能带来的片面性和判断不准确性。随机森林中的每一棵决策树都是独立、无关联的,当对一个新的样本进行判断或预测时,让森林中的每一棵决策树分别进行判断,看看这个样本应该属于哪一类,然后统计哪一类被选择最多,就预测这个样本为哪一类。

1.4 分类模型的评估指标

二分类问题常用准确率、查准率、召回率等指标评估模型的优劣[16],而对于多类别分类问题,有些二分类的评价指标则不适用。

通常使用 Kappa 系数对多类别分类模型进行评估。Kappa 系数是统计学中用于评估一致性的一种方法[17],分类问题的一致性就是模型的预测结果与实际分类结果是否一致。Kappa 系数的取值范围是[-1,1],值越大,则表示模型的分类性能越好。

Kappa 系数的计算公式为:

$$\text{Kappa} = \frac{p_0 - p_e}{1 - p_e} \tag{2}$$

其中,p_0 是每一类正确分类的样本数量之和除以总样本数,也就是总体分类精度。假设每一类的真实样本个数分别为 $a_1, a_2, \cdots, a_c$;而预测出来的每一类的样本个数分别为 $b_1, b_2, \cdots, b_c$;总样本个数为 n,则有:

$$p_e = \frac{a_1 * b_1 + a_2 * b_2 + \cdots + a_c * b_c}{n * n} \tag{3}$$

2 多类别文本分类方法比较方案的设计

2.1 比较对象

本研究在比较对象上考虑了文本特征表示方法和分类算法两个维度。其中,文本特征表示选取了 TF-IDF、词嵌入 CBOW 和词嵌入 SKIP-GRAM 三种方法;分类算法包括 5 种:决策树、SVM、KNN、朴素贝叶斯和随机森林。对不同的文本特征表示方法和分类算法进行组合,构成 15 种分类模型,以这 15 种分类模型为比较对象。

2.2 比较指标

在比较指标上考虑了时间和分类效果。分类效果使用 Kappa 系数来衡量,时间方面包括:(1)文本特征表示的处理时间;(2)分类模型的构建时间与测试样本的预测时间之和。时间均以秒为单位。

2.3 比较流程

在比较流程上考虑了数据规模与比较次数。基本思路是:

(1)在原始数据集中随机采样 N 条数据;

(2)分别使用 TF-IDF、词嵌入 CBOW、词嵌入 SKIP-GRAM 方法构建这 N 条数据的文本特征矩阵,将这 N 条数据按照一定的比例(如 8∶2)拆分成训练集和测试集;

(3)分别使用 SVM、KNN 等不同的分类算法基于不同的文本特征表示构建分类模型,并对测试集进行预测,统计各模型的 Kappa 系数、运行时间等指标;

(4)重复步骤(1)~(3) M 次(如 $M=50$)后,计算在数据规模为 N 条数据时,M 次比较后各比较指标的平均值;

(5)增加数据规模后继续执行步骤(1)~步骤(4),如设定每次增加 200 条数据,即 $N = N + 200$,得到新的数据规模下 M 次比较后各比较指标的平均值;

(6)当数据规模超过了原始数据集的条数后停止比较,综合评估不同数据规模下不同模型的性能。

3 多类别文本分类方法比较实例

3.1 数据集

3.1.1 原始数据集

使用八爪鱼采集器从好奇心日报、新浪网、网易等媒体阅读网站爬取了 3 000 条不同类别的资讯文本,整理成研究所需要的原始数据集,保存到 CSV 格式的文件中。该数据集由分类、标题、正文三个字段组成,如图 2 所示。

类别	标题	正文
商业	木门电商强势来袭	互联网电商一直很火爆,而今年更是达到了最高点,众多企业都纷纷开启电商模式,建材家居企业
时尚	小个子不能穿长款大衣?我不同意!	暖款外套已经无法满足天寒地冻的冬天了,大家都在纷纷入手美丽又保暖的长款大衣or羽绒服。每到
文化	宋朝,不是只有苏黄米蔡!	宋代,是中国文化璀璨斑斓的朝代,涌现了一大批书画诗词名家。宋代书法在中国书法史上占有重要
游戏	全民电竞路线下,王者荣耀让电竞与城市文化完美融合	全民电竞路线下,王者荣耀让电竞与城市文化完美融合作为国民IP,王者荣耀一直希望通过电竞赛事
娱乐	方力申“弃”邓丽欣《宅男总动员》约会赵柯	方力申、赵柯牵手《宅男总动员》方力申“宅男”约会纸条传情赵柯出演电影《宅男总动员》孔雀剩
智能	小冰框架虚拟女友今日开启公测光棍节不再孤独一人	人工智能小冰框架虚拟人类产品线——虚拟女友开启公测。目前,在小米小爱同学、华为快应用、微

图 2 原始数据集示例

其中,文本类别有 6 个:商业、娱乐、游戏、文化、智能和时尚,各类别文本的数据量在原始数据集中大致呈平均分布,数据集适合做多类别文本分类研究。

3.1.2 训练数据集与测试数据集

本实例只研究文本标题的自动分类,因此训练集和测试集只涉及到类别和标题两个字段。如前文所

述,在比较过程中,每次从原始数据集中采样一定规模的数据,将这些数据按照 8∶2 的比例拆分成训练集和测试集。采样规模从 400 条逐渐递增到 3 000 条,步长为 200,并且,同一规模的训练集和测试集进行 50 次建模比较。

3.1.3　原始数据集中“正文”字段的作用

原始数据集中每一条数据的正文都是一个长文本,正文总字数达到了 7 854 428,完全可以将正文内容作为训练词嵌入模型的语料库。

3.2　标题文本的特征表示

3.2.1　TF-IDF 文本特征表示

在 R 语言环境下使用 quanteda 包中的 corpus( )、tokens( )、dfm( )、dfm_tfidf 等函数构建标题的 TF-IDF 文本特征表示模型,主要语法如下:

原始文件<-read. csv(文件名. csv)

标题内容<-corpus(原始文件 $ 标题)

分词<-tokens(标题内容)

分词<-tokens_remove(分词, stopwords(language = " zh" ,source = " misc"))

文档词条矩阵<-dfm(分词)

TF-IDF 文本特征表示<-dfm_tfidf(文档词条矩阵)

以采样 400 条数据为例,得到的标题文本的 TF-IDF 文本特征矩阵如图 3 所示。

```
Document-feature matrix of: 400 documents, 5
 features (99.6% sparse) and 7 docvars.
       features
docs       宜家    线上    开售 智能  百叶窗
  text1 2.60206 2.60206 2.30103    2 2.60206
  text2 0       0       0          0 0
  text3 0       0       0          0 0
  text4 0       0       0          0 0
  text5 0       0       0          0 0
  text6 0       0       0          0 0
[ reached max_ndoc ... 394 more documents ]
```

图 3　标题文本的 TF-IDF 表示矩阵示例

很显然,使用 TF-IDF 进行文本特征表示文档词条矩阵过于庞大并高度稀疏。

3.2.2　基于词嵌入的文本特征表示

使用 R 语言的 word2vec 包构建基于词嵌入的文本表示,主要步骤如下:

步骤 1:词嵌入模型训练文本的分词、去停用词等处理。

如前文所述,本实例将原始数据集中“正文”字段的所有文本作为训练词嵌入模型的语料库。由于 word2vec 算法的输入是词语列表而不是整篇文章,因此首先需要对训练语料库进行分词、去停用词、去符号、去数字等处理,并将分词后的语料文件保存成 CSV 格式文件以备后续训练词嵌入模型使用。

步骤 2:使用语料文件训练词嵌入模型。

使用步骤 1 形成的语料文件和 word2vec 函数生成词嵌入模型。主要语法如下:

词嵌入语料<-read. csv(语料文件. csv)

CBOW 词嵌入模型<- word2vec(x = 词嵌入语料 $ 语料库词条, type = " cbow" , dim = 50, iter = 20, split = " ")

将 word2vec 函数中的参数 type 设定为“skip-gram”则可以训练出 SKIP-GRAM 词嵌入模型,即:

SKIPGRAM 词嵌入模型<- word2vec(x = 词嵌入语料 $ 语料库词条, type = " skip-gram" , dim = 50, iter = 20, split = " ")

步骤 3:使用词嵌入模型对标题文本进行特征表示。

基于步骤 2 训练出的词向量模型,使用 doc2vec 函数将分词后的标题内容表示成向量模型,即将每个标题内容映射成 50 维实数向量。主要语法如下:

文档 ID<-seq(1:采样条数))

数据框<- data. frame(doc_id = 文档 ID, text = 标题文本分词后的词表, stringsAsFactors = FALSE)

基于 CBOW 词嵌入的文本特征表示<-doc2vec(CBOW 词嵌入模型,数据框,type = " embedding")

基于 SKIP-GRAM 词嵌入的文本特征表示<-doc2vec(SKIPGRAM 词嵌入模型, 数据框, type = " embedding")

通过上述过程,将每一个标题文本映射成 50 维的实数向量。

3.3　文本分类模型的构建及性能评估

对于本研究所涉及的 SVM、KNN、决策树、朴素贝叶斯、随机森林五种分类算法,在 R 语言环境下,使用 party、e1071、randomForest 等包中提供的函数进行文本分类模型的构建。

以使用 randomForest 包中的随机森林算法构建基于不同特征表示的分类模型为例:

(1)基于 TF-IDF 的文本特征表示。

TFIDF 分类模型<-randomForest(类别 ~ . , TFIDF 特征表示的训练数据集, ntree = 30, na. action = na. roughfix)

TFIDF 预测结果<-predict(TFIDF 分类模型, TFIDF 特征表示的测试数据集, proximity = TRUE)

(2)基于词嵌入 CBOW 的文本特征表示。

CBOW 分类模型<-randomForest(类别 ~ . , CBOW 特征表示的训练数据集, ntree = 30, na. action = na. roughfix)

CBOW 预测结果<- predict(CBOW 分类模型, CBOW 特征表示的测试数据集, proximity = TRUE)

(3)基于词嵌入 SKIP-GRAM 的文本特征表示。

SKIP－GRAM 分类模型<－randomForest(类别~.,SKIP－GRAM 特征表示的训练数据集,ntree=30,na. action=na. roughfix)

SKIP－GRAM 预测结果<－ predict(SKIP－GRAM 分类模型,SKIP－GRAM 特征表示的测试数据集,proximity=TRUE)

在模型构建及对测试数据集进行预测的过程中统计运行时间,并且在预测之后构建预测值与真实值的混淆矩阵,使用 VCD 包中的 Kappa 函数基于混淆矩阵计算模型的 Kappa 系数,衡量模型的分类效果。

3.4 模型的比较结果

3.4.1 文本特征表示处理时间的比较

在不同的数据规模下,使用 TF－IDF、词嵌入 CBOW 和词嵌入 SKIP－GRAM 三种方法对文本进行特征表示的处理时间变化如图 4 所示。

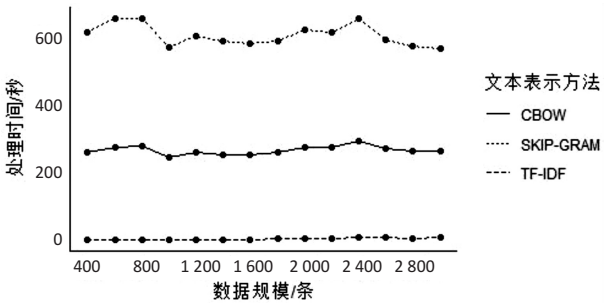

图 4 不同数据规模下使用不同方法进行文本特征表示的处理时间

从图 4 可以看出:

在相同的数据规模下,词嵌入的文本特征表示处理时间都远远超过 TF－IDF,这是因为词嵌入需要对大量的语料库进行训练,而在两种词嵌入方法中,SKIP－GRAM 比 CBOW 的训练时间更长(大约是 2.5 倍)。

三种特征表示的处理时间与数据规模的相关系数如表 1 所示。

表 1 文本表示处理时间与数据规模的相关性

文本表示方法	处理时间与数据规模的相关系数
TF－IDF	0.934 792 149
CBOW	0.225 552 518
SKIP－GRAM	－0.376 843 38

从表 1 可以看出:

(1)TF－IDF 文本特征表示的处理时间与数据规模高度正相关,采样数据越多,处理的词条数越多,TF－IDF 文本特征表示的处理时间越长;

(2)两种词嵌入特征表示的处理时间与所处理数据的数据规模之间的相关性不强。

3.4.2 模型构建与预测时间比较

15 种模型在不同数据规模下运行时间的变化如图 5 所示。由于使用 TF－IDF 进行文本特征表示的模型与使用词嵌入进行文本特征表示的模型在运行时间上数值范围相差极大,所以在图 5 中用上下两幅图来阐释,上图表示使用 TF－IDF 进行文本特征表示的模型,下图表示使用词嵌入进行文本特征表示的模型。

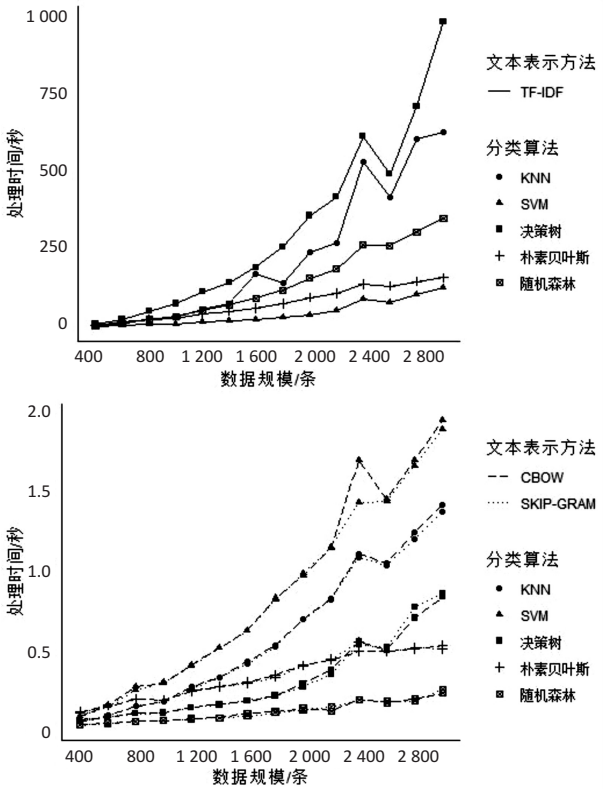

图 5 不同数据规模下不同模型的分类建模与预测时间

从图 5 可以看出:

(1)15 种模型的运行时间均与数据规模高度正相关,相关系数如表 2 所示。

表 2 模型的运行时间与数据规模的相关性

模型	运行时间与数据规模的相关系数
朴素贝叶斯+CBOW	0.991 722 385
朴素贝叶斯+TF－IDF	0.990 056 371
SVM+SKIP－GRAM	0.986 382 834
KNN+SKIP－GRAM	0.979 314 831
KNN+CBOW	0.978 604 013
SVM+CBOW	0.974 602 752
随机森林+TF－IDF	0.974 528 998
朴素贝叶斯+SKIP－GRAM	0.974 377 655
随机森林+CBOW	0.971 259 836
随机森林+SKIP－GRAM	0.964 911 435
决策树+TF－IDF	0.941 361 321
KNN+TF－IDF	0.939 070 513
SVM+TF－IDF	0.936 095 281
决策树+CBOW	0.930 914 63
决策树+SKIP－GRAM	0.918 606 081

(2)在相同数据规模、相同的文本分类算法下,文本特征表示使用 TF-IDF 的模型运行时间远远超过文本特征表示使用词嵌入模型的运行时间。

(3)综合来看,在相同的数据规模下,随机森林+CBOW 模型的运行时间最短;而决策树+TF-IDF 模型的运行时间最长。最短时间与最长时间的线性拟合关系如图 6 所示。

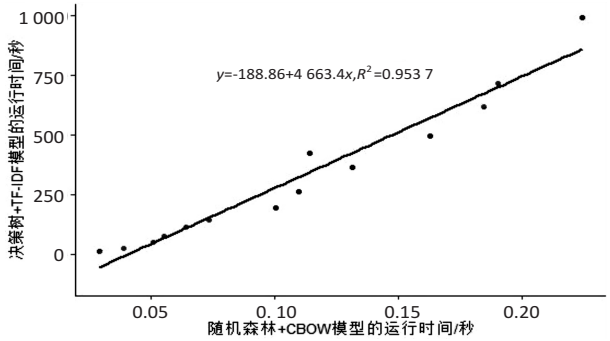

图 6　随机森林+CBOW 及决策树+TF-IDF 模型运行时间的线性拟合

3.4.3　模型的分类效果比较

文本多分类模型的分类效果使用 Kappa 系数来衡量,15 种模型在不同数据规模下 Kappa 系数的变化如图 7 所示。

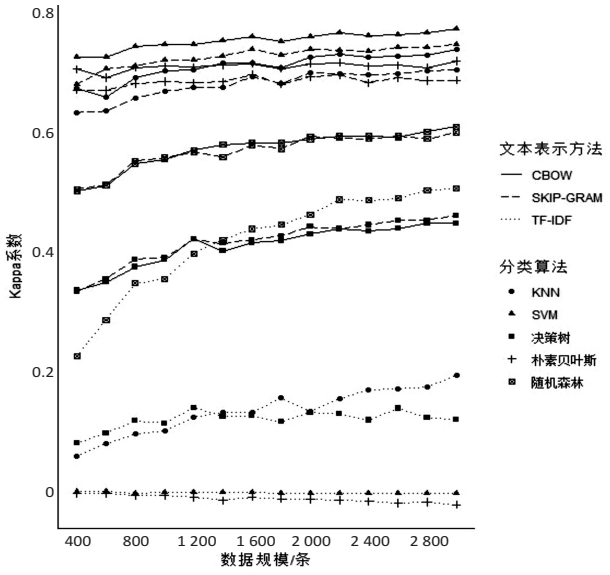

图 7　不同数据规模下不同模型的分类效果

从图 7 可以看出:

(1)在本研究的任何一种数据规模下,SVM+CBOW 模型的分类效果都是最好的;而朴素贝叶斯+TF-IDF 模型的分类效果则最差。

(2)在相同的数据规模、相同的分类算法下,文本表示使用 TF-IDF 方法的模型分类效果都是最差的;使用词嵌入方法的分类模型的 Kappa 系数要比使用 TF-IDF 的模型的 Kappa 系数大很多;两种词嵌入模型的 Kappa 系数相差不大,总体来说,CBOW 模型的分类效果略优于 SKIP-GRAM 模型。

(3)随机森林作为集成算法,容易给人造成的误解是:其性能一定比单一算法要好。但比较结果发现,在本研究中,当使用词嵌入进行文本特征表示时,随机森林的分类效果虽然比单一决策树的分类效果要好,但却比 SVM、KNN、朴素贝叶斯的分类效果差;当使用 TF-IDF 进行文本特征表示时,随机森林的分类效果最好,然后依次是决策树、KNN、SVM 和朴素贝叶斯。这说明:随机森林在高维度、大规模数据集的分类处理上具有一定的优势,但对于少量和低维数据集的分类不一定可以得到很好的分类效果。

15 种模型的分类效果与数据规模的相关性如表 3 所示。

表 3　模型的分类效果与数据规模的相关性

模型	分类效果与数据规模的相关系数
KNN+TF-IDF	0.968 925
随机森林+TF-IDF	0.940 404
决策树+SKIP-GRAM	0.936 07
决策树+CBOW	0.926 375
KNN+SKIP-GRAM	0.922 379
SVM+CBOW	0.920 294
随机森林+CBOW	0.908 727
KNN+CBOW	0.905 538
随机森林+SKIP-GRAM	0.897 754
SVM+SKIP-GRAM	0.887 613
朴素贝叶斯+SKIP-GRAM	0.631 486
朴素贝叶斯+CBOW	0.596 146
决策树+TF-IDF	0.579 597
SVM+TF-IDF	-0.751 88
朴素贝叶斯+TF-IDF	-0.951 23

从表 3 可以看出:

(1)KNN 算法和随机森林算法与高维的 TF-IDF 文本表示方法组合时,数据规模越大分类效果越好,说明这两种算法适合对数据量大、高维的数据集进行分类处理。

(2)决策树算法与低维的词嵌入文本表示方法组合时,分类数据量越大分类效果越好,说明决策树方法适合对大量的低维数据进行分类处理。

(3)朴素贝叶斯和 SVM 算法明显不适合对高维数据进行处理,当使用 TF-IDF 进行文本表示时,分类数据量越大,这两种算法的分类效果越差。

4　结束语

本研究综合考虑了数据规模、数据维度(文本表

示方法)、分类算法三方面,设计了多类别文本分类方法比较方案,从时间和分类效果两个维度评估分类模型的性能。综合评估后认为,对于多类别文本分类问题:

(1)文本特征表示不建议使用 TF-IDF 方法。使用 TF-IDF 方法,尽管在前期文本特征表示的处理时间上有一定的优势,但是由于文本特征矩阵过于稀疏和庞大,导致分类模型的运行时间过长、分类效果亦极不理想。

(2)在两种 word2vec 词嵌入算法中,建议选择 CBOW 方法,该方法不仅在文本特征表示阶段具有明显的时间优势,而且在建模阶段,CBOW 与朴素贝叶斯、SVM、KNN 算法组合的模型分类效果均非常理想。

(3)在分类算法的选择上,当数据规模不是很大时,不建议选择随机森林等集成算法,随机森林算法的优势体现在对高维数据的处理上,其与词嵌入文本表示方法组合未必能达到非常理想的分类效果。

参考文献:

[1] 叶雪梅,毛雪岷,夏锦春,等. 文本分类 TF-IDF 算法的改进研究[J]. 计算机工程与应用,2019,55(2):104-109.

[2] 刘 宇,崔燕红,郭师光,等. 聊天机器人:入门、进阶与实战[M]. 北京:机械工业出版社,2019.

[3] 汪 静,罗 浪,王德强. 基于 Word2Vec 的中文短文本分类问题研究[J]. 计算机系统应用,2018,27(5):209-215.

[4] 余 冲,李 晶,孙旭东,等. 基于词嵌入与概率主题模型的社会媒体话题识别[J]. 计算机工程,2017,43(12):184-191.

[5] 冯 冲,石 戈,郭宇航,等. 基于词向量语义分类的微博实体链接方法[J]. 自动化学报,2016,42(6):915-922.

[6] NAILI M,CHAIBI A H,GHEZALA H H B. Comparative study of word embedding methods in topic segmentation[J]. Procedia Computer Science,2017,112:340-349.

[7] 高明霞,李经纬. 基于 word2vec 词模型的中文短文本分类方法[J]. 山东大学学报:工学版,2019,49(2):34-41.

[8] 张 棪,曹 健. 面向大数据分析的决策树算法[J]. 计算机科学,2016,43(S1):374-379.

[9] THANGARAJ M, SIVAKAMI M. Text classification techniques:a literature review[J]. Interdisciplinary Journal of Information,Knowledge & Management,2018,13:117-135.

[10] 苏慧婧,群 诺,贾宏云. 基于 KNN 模型的藏文文本分类研究与实现[J]. 高原科学研究,2019,3(2):88-92.

[11] 陈志云,商 月,钱冬明. 基于知识图谱的智能答疑系统研究[J]. 计算机应用与软件,2018,35(2):178-182.

[12] 金 权,华 锋,杨永增. 基于 SVM 的海浪要素预测试验研究[J]. 海洋科学进展,2019,37(2):199-209.

[13] GOUDJIL M,KOUDIL M,BEDDA M,et al. A novel active learning method using SVM for text classification[J]. International Journal of Automation and Computing, 2018, 15(3):290-298.

[14] 文孟飞,刘伟荣,叶 征. 基于自动聚类和集成学习的网络教学形成性评价方法[J]. 中国电化教育,2018(3):74-82.

[15] SHI L J,LU J. Automatic measurement method for maize ear development degree based on random forest[J]. Transactions of the Chinese Society for Agricultural Machinery,2017,48(1):169-174.

[16] KUMARI R,SRIVASTAVA S K. Machine learning:a review on binary classification[J]. International Journal of Computer Applications,2017,160(7):11-15.

[17] 徐树良,王俊红. 基于 Kappa 系数的数据流分类算法[J]. 计算机科学,2016,43(12):173-178.