# 商品名称短文本快速有效分类的多基模型框架

沈雅婷 左志新*
(南京理工大学紫金学院 江苏 南京 210023)

**摘 要** 提出一种适用于短文本分类的多基模型框架 Bagging_fastText(B_f)。它是一种基于自举汇聚法的快速文本分类算法的框架。以 fastText 为基模型,运用集成学习思想,设置最优超参数并训练出多个基模型组成多基模型,再通过投票机制获取最终类别。对商品名称短文本分类的实验结果表明,提出的 B_f 比 fastText、朴素贝叶斯传统文本分类算法、文本卷积神经网络(TextCNN)算法的分类效果更优。

**关键词** 多基模型框架 fastText 文本分类 NLP

**中图分类号** TP39 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2021.02.031

## MULTI-BASE MODEL FRAMEWORK FOR FAST AND EFFECTIVE CLASSIFICATION OF SHORT TEXT OF COMMODITY NAMES

Shen Yating Zuo Zhixin*
(*ZiJin College*, *Nanjing University of Science and Technology*, *Nanjing 210023*, *Jiangsu*, *China*)

**Abstract** This paper proposes a multi-base model framework for short text classification, Bagging_fastText(B_f). It is a framework of fast text classification algorithm based on Bootstrap aggregating method. It used fastText as the base model, used ensemble Learning idea, set optimal hyperparameters and trained multiple base models to form multi-base model, and then the final classification was obtained by voting mechanism. The experimental results of short text classification of product names show that the proposed B_f has better classification effect than the fastText, naive Bayesian traditional text classification algorithm, and TextCNN(text convolutional neural network) algorithm.

**Keywords** Multi-base model framework FastText Text classification NLP

## 0 引 言

分类一直是自然语言处理(NLP)领域的重点问题,被广泛地应用到生活之中。随着中国互联网信息技术行业的快速发展以及大数据时代的到来,电子文本数据呈现井喷式增长,例如微博、商品评论、商品名称等短文本数据占比最大。合理地对其进行分类,便于用户快速查找所需信息或商品,也便于商家及时发现问题和掌握用户需求,提高用户满意度。针对产生的大量短文本数据,人工分类存在速度慢,成本高等问题,所以需要机器去代替人工完成分类。

文本分类问题是 NLP 领域中一个非常经典的问题,最早可以追溯到 20 世纪 50 年代,早期主要通过知识工程,手工定义规则来分类,不仅浪费时间和人力,而且使用范围和准确度都十分有限。随着 20 世纪 90 年代互联网在线文本的涌现和机器学习的兴起,研究者重新开始对文本分类的研究,逐渐将文本分类问题拆分成特征工程和分类器两个部分,即基于传统机器学习的文本分类。其中特征工程包括文本预处理、特征提取和文本表示等,分类器基本都是统计分类的方法,如 K 近邻、朴素贝叶斯[1]、决策树、支持向量机等。以上方法相比早期的方法有着更好的分类效果,但是文本表示的特征表达能力较弱,还需要人工进行特征工程且非常费时费力,成本很高。随着计算能力提升、成本下降、海量大数据支持和人工神经网络兴起,基于

收稿日期:2019-07-24。江苏省高校自然科学研究项目(19KJB520039);南京理工大学紫金学院科学研究项目(2019ZRKX0401008)。**沈雅婷**,讲师,主研领域:机器学习,深度学习。**左志新**,本科生。

人工神经网络的深度学习方法逐渐成为主流的研究方向,深度学习方法利用人工神经网络的网络结构自动获取特征表达的能力解决文本分类的文本表示问题[2],避免了繁杂的人工特征工程,即端到端地解决问题。在文本分类中,常用的神经网络模型有 TextCNN[3-4]、TextRNN、fastText[5]等。

直至目前,文本分类在工业界和学术界已经积累了很多方法,主要分为基于传统机器学习、基于深度学习两种文本分类实现方法。基于深度学习的文本分类方法比基于传统机器学习的文本分类方法准确度高,而常规深度学习的文本分类方法中的神经网络训练时间较长。短文本分类的本质还是文本分类,可以直接使用这些方法,但是短文本存在长度短、特征稀疏、表述不规范等特点导致分类性能明显下降。目前针对短文本的特点,主要有通过外部语料库构建词向量、利用网络资源构建专门的类别词库对特定的短文本进行扩展等方法,以提高短文本的分类性能。虽然这些方法降低了短文本的稀疏性,提高了分类准确率,但是获取覆盖所有类别的外部语料库是很困难的。并且在大数据的时代背景下,短文本数据的流通变得越来越快,已有的文本分类方法无法达到既准确率高又训练速度快,这就限制了其在大数据背景下的广泛使用,渐渐不能满足时代对信息处理的高速要求。因此,为了满足时代要求,迫切需要寻求一种新的方法以实现对短文本又好又快地自动分类。

本文提出一种新的模型框架(B_f),以快速文本分类算法(fastText)作为基模型,借鉴自举汇聚法(Bagging)集成算法基本思想,构建多基模型的方法对文本进行分类。通过实验证实提出的方法在文本分类中较独立基模型具有不错的效果,较基于机器学习的传统文本分类算法和深度学习分类方法在准确度与效率上有明显优势。

## 1 相关知识

### 1.1 fastText 模型

fastText 是一个基于浅层神经网络、架构简单的快速文本分类器,由 Facebook 在 2016 年开源。其优点是可以使用标准多核 CPU 在 10 分钟内训练超过 10 亿个词,比常规深度学习模型快几百倍,在 5 分钟内可以对超过 30 万个类别的 50 万个句子完成分类。其通过使用 $n$-gram($N$ 元模子)特征来缩小线性规模,其分类准确度能够与常规深度学习模型保持相当,但训练时间却大大短于常规深度模型,分类速度也比深度模型快很多。

fastText 由输入层、隐藏层和输出层三层构成,其模型结构如图 1 所示。

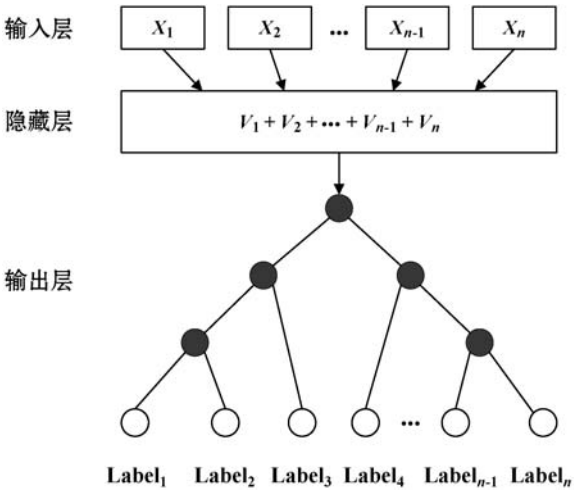


图 1 fastText 模型结构

图 1 中:$X_n$ 表示第 $n$ 个单词以及字符的 $n$-gram 特征的 one-hot 表示;$V_n$ 表示第 $n$ 个单词以及字符的 $n$-gram 特征密集向量表示;$\text{Label}_n$ 表示第 $n$ 个标签。

fastText 是将已经分词后的文本作为输入,输出该文本属于不同类别的概率。使用文本中的词和词组构成特征向量,通过线性变换,将特征向量映射到隐藏层,然后构建层次 Softmax 分类器根据类别的权重和模型参数使用 Huffman 编码对标签进行编码,将 Huffman 树作为输出[6-9],Huffman 树的叶子节点即为标签。

当数据集的类别很多时,线性分类器的计算会变得很昂贵。为了降低 Softmax 层的计算复杂度,fastText 使用了一个基于 Huffman 编码树的分层 Softmax。在这个 Huffman 树中,每个叶子节点即代表一个标签。利用了类别不平衡的这个事实,将每个类别出现的频率作为权重,使用 Huffman 算法构建 Huffman 树,出现频率高的类别比出现频率低的类别深度要小,使得计算效率更高。

常用的特征有词袋(Bag-of-words,BoW)模型和 $n$-gram 特征。其中词袋模型不考虑词之间的顺序,但是对于很多分类问题而言,词序十分重要,如果词序不同,文本含义可能截然相反,但是直接考虑顺序的计算成本又很高昂,而 $n$-gram 考虑了局部词序。因此,fastText 使用 $n$-gram 特征,通过向量表示单词 $n$-gram 来将局部词序考虑在内,过滤掉低频的 $n$-gram,从而提高效率[10]。

### 1.2 Bagging 集成学习算法

Bagging[11]是通过多个模型相结合降低泛化误差的技术,把多个不同的个体分类器集成为一个分类器

的集成学习方法,主要思想是将训练数据有放回地抽样训练多个不同模型,然后将所有模型对测试样例的表决输出。由于 Bagging 集成学习算法的个体分类器之间没有强依赖关系,从而可以并行,可使用分布式计算进一步提高算法的效率。

### 1.3　其他文本分类方法

1) TextCNN 模型。CNN 是一种前馈神经网络,广泛应用于模式识别、图像处理等领域,是深度学习的代表算法之一。2014 年,纽约大学 Yoon Kim 将 CNN 应用在文本分类上提出 TextCNN 模型,一个简单且具有少量超参数调整的 CNN,可根据具体任务进行微调进一步提高性能。对矩阵化的文本进行卷积和最大池化后,再通过全连接层的 Softmax 进行结果输出。由于其结构简单、效果好,在文本分类任务上有着广泛的应用。

2) 朴素贝叶斯模型。贝叶斯算法是以贝叶斯原理为基础,使用数理统计的知识对样本数据集进行分类。贝叶斯分类算法的误判率很低,在数据集较大的情况下表现出较高的准确率,同时算法本身也比较简单。朴素贝叶斯算法是在贝叶斯算法的基础上进行了相应的简化,即假设给定目标值时属性之间相互条件独立,也就是特征向量中一个特征的取值并不影响其他特征的取值,虽然在一定程度上降低了贝叶斯算法的分类效果,但是由于其实现简单且表现惊人,成为应用最为广泛的分类模型之一。

## 2　框架设计

B_f 使用 fastText 模型作为基模型,借鉴 Bagging 集成算法的基本思想,挑选 fastText 的两组最优超参数。15 次打乱预处理后的训练数据作为训练样本集分别进行训练,最后构建由 15 个产生的基模型组成的多基模型,结合少数服从多数的投票机制才能对预处理后的测试样本数据进行标签预测,B_f 的总体流程如图 2 所示。

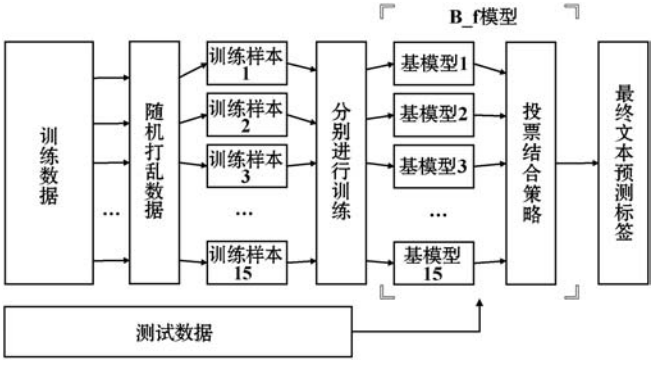


图 2　B_f 总体流程

### 2.1　预处理

在使用数据之前,需要对原始文本数据进行预处理工作。主要流程有分词和去停用词等,步骤详述如下:

1) 中文分词。中文与英文不同,英文是以词为单位,词与词之间用空格分隔,而中文是以字为单位,需要使用相关分词工具将中文汉字序列分割成词并用空格分隔[12]。中文分词算法可以分为三类:(1) 字符串匹配算法,其核心思想是词典匹配完成词语切分;(2) 基于理解的分词算法,其基本思想是在分词的同时进行句法、语义分析,因为中文语言的复杂性,目前基于理解的分词系统还处于试验阶段;(3) 基于统计的分词算法,其主要思想是将每个词看作是由字组成的,如果相连的字在不同文本中出现的频率越多,则证明这段字越有可能是一个词。目前 Python 常用的分词工具有 jieba 分词、THULAC(一个高效的中文词法分析工具包)等。因此本文选取了具有分词速度快、准确率高和使用简单等特点的 jieba 分词作为本文使用的分词工具。部分文本使用 jieba 分词样例如表 1 所示。

**表 1　部分文本使用 jieba 分词样例**

| 原文本 | 分词后文本 |
|---|---|
| 腾讯 QQ 黄钻三个月 QQ 黄钻 3 个月季卡官方自动充值可查时间可续费 | 腾讯 QQ 黄钻 三个月 QQ 黄钻 3 个 月季卡 官方 自动 充值 可 查 时间 可 续费 |
| 断码清仓特价内增高甜美时尚真皮单鞋 2016 春秋季新款铆钉大码休闲坡跟鞋平底鞋圆头舒适女式鞋 | 断码 清仓 特价 内 增高 甜美 时尚 真皮 单鞋 2016 春秋季 新款 铆钉 大码 休闲 坡 跟 鞋 平底鞋 圆头 舒适 女式 鞋 |
| 欧玛奴高档全包汽车坐垫透气冰丝拼皮座垫四季通用夏天专用皮坐垫冰丝椅垫车垫套三菱斯巴鲁斯柯达 | 欧玛奴 高档 全包 汽车坐垫 透气 冰丝 拼皮 座垫 四季 通用 夏天 专用 皮 坐垫 冰丝 椅垫 车垫 套 三菱 斯巴鲁 斯柯达 |
| 虎牌双保险防盗电子密码双保险保险箱全钢制造全钢保管箱 BGX-A1/D-50 土豪金 | 虎牌 双保险 防盗 电子 密码 双保险 保险箱 全钢 制造 全钢 保管箱 BGX-A1/D-50 土豪金 |

(2) 去停用词。去停用词为了将文本中一些出现频率高、无实际意义、对有效信息噪音干扰的词去掉,如“的”“是”“和”等,并且可以节省计算机的存储与计算资源[13]。本文使用“哈工大停用词表”、“四川大学机器智能实验室停用词库”和“百度停用词表”相整合的停用词表作为中文停用词表对文本进行过滤。

### 2.2　超参数调优

超参数[14-15]是机器学习以及深度学习模型内的框架参数,是在学习之前设置的参数,而不是通过训练

得到的。通常,需要对超参数进行调优,给学习机选择一组最优超参数,以提高学习的性能和效果,是一项繁琐但至关重要的任务。通常需要手动设定,不断试错调试,需要大量的专家经验;也可以通过贝叶斯优化算法[16-17]等自动的优化模型进行调优。

由于 $n$-gram 超参数是 fastText 模型一个重要的超参数,能够影响模型的时间效率以及分类精度,所以将 $n$-gram 超参数设置为一个固定的值,再进行调优可以大幅度提高超参数调优的进度。通过多次手动调优的实验发现,对于商品名称而言,$n$-gram 超参数设置为 1 或 2 时,模型的时间效率以及分类精度最好,研究者需要根据具体数据进行微调。

将预处理后的商品名称训练数据划分为训练集和验证集,然后分别将 fastText 模型的 $n$-gram 超参数设定为 1 和 2,进行超参数调优,获得两组最优超参数。

## 2.3 构建多基模型

B_f 共由 15 个 fastText 基模型组成,其中 7 个由 $n$-gram 超参数为 1 的最优超参数组作为超参数训练得到,另外 7 个由 $n$-gram 超参数为 2 的最优超参数组作为超参数训练得,最后 1 个基模型是在这两组最优超参数组中随机抽取一组最为超参数训练得。如图 3 所示。当对文本进行预测时采用投票机制[18]融合,得到最终预测标签。多基模型如图 2 所示。

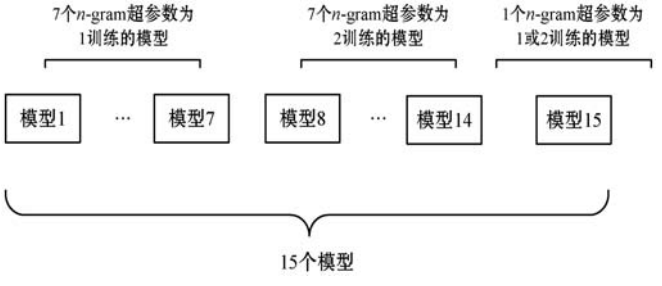


图 3 多基模型示意图

每个基模型的训练数据都是由训练集随机打乱而来。对于神经网络来说当训练集较大时,训练集前面的样本对模型权重的影响会随着训练变小,通过多次的打乱达到一种综合的目的。多基模型相对于独立模型而言,对容易产生歧义的样本分类更加有效,单个模型对不同类别的样本分类具有偏向性,实现效果有限,使用多个模型组合能够提高模型的泛化能力[19-20]。

通常这种做法是不鼓励使用的,因为它是以增加计算和存储作为代价。但是 fastText 的优点在于其结构简单、时效性好,分类效果也相对令人满意。本文模型在使用 15 个基模型的情况下,其时效也可以令人满意,对比没有 GPU 加速的其他深度模型在时效以及准确率上都有着十分明显的优势。

## 2.4 算法描述

输入:训练集 $D$;fastText 算法 A。

输出:B_f 分类器 $C(X)$。

**Step 1** 对训练集 $D$ 进行预处理;创建预处理后的训练集 $D_1$。

**Step 2** 使用预处理后的训练集 $D_1$ 进行 fastText 算法 A 超参数调优;创建最优超参数组 $P_1$;创建最优超参数组 $P_2$。

**Step 3** for $i=1$ to 7

**Step 4** 将预处理后的训练集 $D_1$ 随机打乱;创建样本集 $D_i$。

**Step 5** 用样本集 $D_i$ 和使用最优超参数组 $P_1$ 作为超参数的 fastText 算法 A 训练,得到基分类器 $c_i(x)$。

**Step 6** end for

**Step 7** for $i=7$ to 14

**Step 8** 将预处理后的训练集 $D_1$ 随机打乱;创建样本集 $D_i$。

**Step 9** 用样本集 $D_i$ 和使用最优超参数组 $P_2$ 作为超参数的 fastText 算法 A 训练,得到基分类器 $c_i(x)$。

**Step 10** end for

**Step 11** 将预处理后的训练集 $D_1$ 随机打乱;创建样本集 $D_{15}$。

**Step 12** 用样本集 $D_{15}$ 和随机使用最优超参数组 $P_1$ 或 $P_2$ 作为超参数的 fastText 算法 A 训练,得到基分类器 $c_{15}(x)$。

**Step 13** 输出 B_f 分类器

$$C(X) = \arg\max \sum_{i}^{15} c_i(x) \tag{1}$$

使用 B_f 分类器 $C(X)$ 对未知样本 $x$ 分类:

未知样本 $x$ 分类时,每个分类器 $c_i(x)$ 得出一个分类结果,15 个分类器投票,得票最多的类别即为未知样本 $x$ 的分类结果,并输出分类结果:

$$C(x) = \arg\max \sum_{i}^{15} c_i(x) \tag{2}$$

# 3 实 验

## 3.1 实验数据与实验环境

1)实验数据。实验使用浪潮卓数大数据产业发展有限公司提供的网络零售平台商品数据,其中商品名与标签来源于网络。选用其中已标记标签的数据作为实验数据,包含本地生活——游戏充值——QQ 充值、本地生活——游戏充值——游戏点卡、宠物生活——宠物零食——磨牙/洁齿等 1 260 个类别。共有 50 万条数据,本文将分别取数据的 100%、50% 和 1% 作为实验数据,如表 2 所示,其中类别数目 $c$ 随着数据规模增大而递增。实验全部通过十折交叉验证[21]方式进行,使获得的数据真实有效。

**表 2　不同规模下的数据集对比表**

| 数据规模 | 样本数目 | 类别数目 $c$ | 平均长度 |
|---|---|---|---|
| 100% | 50 万 | 1 260 | 43 |
| 50% | 25 万 | 436 | 44 |
| 1% | 0.5 万 | 33 | 38 |

2）实验环境。硬件环境平台:MacBook Pro,处理器 2.6 GHz Intel Core i7,内存 16 GB 2 400 MHz DDR4,macOS Mojave 操作系统。软件环境平台:Python 3.7,scikit-learn,TensorFlow。

## 3.2　衡量方法

本文采用准确率(Accuracy)、精确率(Precision)、召回率(Recall)和综合评价指标(F1-Measure)作为评估指标。准确率(Accuracy)计算式为:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

精确率(Precision)的计算式为:

$$P = \frac{TP}{TP + FP} \tag{4}$$

召回率(Recall)的计算式为:

$$R = \frac{TP}{TP + FN} \tag{5}$$

综合评价指标(F1-Measure)的计算式为:

$$F1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{6}$$

式中:$TP$ 为真正例;$TN$ 为真负例;$FP$ 为假正例;$FN$ 为假负例。

使用十折交叉验证得到最终的准确率、精确率、召回率和综合评价指标,以减小实验结果的误差负例。

## 3.3　结果分析

对于文本分类任务,TextRNN 和 TextCNN 是深度学习中最常见的两大类模型,由于 TextRNN 与 TextCNN 分类效果相差不大,TextCNN 擅长捕获更短的序列信息,TextRNN 擅长捕获更长的序列信息且训练成本更大。本实验使用的商品名称数据集,商品名称描述相对较短,特征词较集中,所以略去 TextRNN,使用 TextCNN 作为对比模型。同时,作为传统文本分类算法的朴素贝叶斯,也较适合处理此类描述相对较短、特征词较集中的文本分类问题。本次实验采用对比分析,在预处理后的实验数据的 100%、50% 和 1% 上进行实验。将 B_f 与单个 fastText 模型、TextCNN 模型以及朴素贝叶斯模型进行对比实验,并采用十折交叉验证方法对这四种算法分别训练十次,将每次训练的输出结果保留,并将十次输出结果取平均值得到四种模型结果的准确率、精确率、召回率和综合评价指标对比如表 3 – 表 6 所示,其中最优值加粗表示。

**表 3　四种模型结果准确率对比**

| 数据规模 | TextCNN | fastText | B_f | N_Bayes |
|---|---|---|---|---|
| 100% | 0.773 2 | 0.826 1 | **0.866 2** | 0.764 4 |
| 50% | 0.863 5 | 0.883 8 | **0.911 8** | 0.840 5 |
| 1% | **0.906 0** | 0.887 8 | 0.892 0 | 0.856 0 |

**表 4　四种模型结果精确率对比**

| 数据规模 | TextCNN | fastText | B_f | N_Bayes |
|---|---|---|---|---|
| 100% | 0.773 6 | 0.8436 | **0.884 7** | 0.755 8 |
| 50% | 0.867 0 | 0.8911 | **0.919 9** | 0.839 4 |
| 1% | **0.915 8** | 0.893 3 | 0.899 1 | 0.860 0 |

**表 5　四种模型结果召回率对比**

| 数据规模 | TextCNN | fastText | B_f | N_Bayes |
|---|---|---|---|---|
| 100% | 0.773 2 | 0.826 1 | **0.866 2** | 0.764 4 |
| 50% | 0.863 5 | 0.883 8 | **0.911 8** | 0.840 5 |
| 1% | **0.906 0** | 0.887 8 | 0.892 0 | 0.856 0 |

**表 6　四种模型结果综合评价指标对比**

| 数据规模 | TextCNN | fastText | B_f | N_Bayes |
|---|---|---|---|---|
| 100% | 0.768 7 | 0.832 0 | **0.872 2** | 0.750 8 |
| 50% | 0.862 6 | 0.885 8 | **0.914 1** | 0.834 4 |
| 1% | **0.907 6** | 0.888 4 | 0.892 3 | 0.852 5 |

可以看出,本文模型在使用的数据规模为 100% 时预测精确率高达 88.47%,其准确率也达到了 86.62%,综合评价指标为 87.22%。当数据规模为 50% 时,本文模型各项指标依然领先与另外三个模型。当数据规模为 1% 时,本文模型各项指标略微低于 TextCNN。不难看出本文模型对于单个 fastText 模型有较大的提升,对比 TextCNN 模型在数据规模较大时也有着明显的优势,对比朴素贝叶斯模型同样有着明显优势。因此可得,本文模型相比于单个 fastText 模型、TextCNN 模型、朴素贝叶斯模型较为理想,达到实验目的。

作为传统文本分类算法的朴素贝叶斯在训练时只需计算概率,并不需要复杂的矩阵计算或者迭代优化,因此朴素贝叶斯模型不作为训练时间参考模型。三种模型一次的训练时长如表 7 所示。不难看出,TextCNN 模型由于结构相对复杂,通常需要很长的训练时间,训练成本较高,在数据规模较大的情况下此问题更加明显。fastText 模型由于结构简单取得了很好的效果,本文模型由于采用 fastText 模型为基模型,也取得相对令

人满意的效果。

表 7 三种模型训练时间对比 s

| 数据规模 | TextCNN | fastText | B_f |
|---|---|---|---|
| 100% | 3 432.09 | **63.07** | 472.68 |
| 50% | 1 604.46 | **34.15** | 244.54 |
| 1% | 24.58 | **4.24** | 29.56 |

## 4 结 语

本文在 fastText 模型的基础上构建了多基模型框架 B_f。它比单个 fastText 模型具有更高的分类准确率;比 TextCNN 等深度学习模型在数据量大的情况下有着更短的训练时长和更高的准确率;比朴素贝叶斯等传统文本分类模型也具有更好的分类效果。能够有效地处理商品名称描述的文本分类问题,同时普遍适用于大规模数据的有监督文本分类问题。

鉴于本文提出的多基模型框架 B_f 的基模型之间没有强依赖关系,可以并行实现。下一步将考虑扩大模型规模,进行在分布式环境下的研究,以进一步提高文本分类的准确率和时间效率。

## 参 考 文 献


[1] 宁可,孙同晶,赵浩强. 基于属性关联的朴素贝叶斯分类算法[J]. 计算机工程,2018,44(6):24 - 29.

[2] 张朝晖,陆玉昌,张钹. 利用神经网络发现分类规则[J]. 计算机学报,1999,22(1):108 - 112.

[3] Kim Y. Convolutional neural networks for sentence classification[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing,2014: 1746 - 1751.

[4] Zhang X, Zhao, J B, LeCunY. Character-level convolutional networks for text classification[C]//International Conference on Neural Information Processing Systems. MIT Press, 2015.

[5] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,2017.

[6] 王艺杰. 基于 Fasttext 的防控目标分类实现[J]. 中国公共安全(学术版),2018(1):29 - 32.

[7] 代令令. 基于 fastText 的问答系统用户意图识别与关键词抽取研究[D]. 南宁:广西大学,2018.

[8] 代令令,蒋侃. 基于 fastText 的中文文本分类[J]. 计算机与现代化,2018(5):35 - 40,85.

[9] 徐华韫,龚泽阳,何正杰,等. 基于深度学习的话题流行度预测[J]. 信息与电脑(理论版),2018(21):57 - 59.

[10] Joulin A, Grave E, Bojanowski P, et al. Bag of tricks for efficient text classification[EB]. arXiv:1607.01759,2016.

[11] 周钢,郭福亮. 集成学习方法研究[J]. 计算技术与自动化,2018,37(4):148 - 153.

[12] 王乔乐. 中文分词和词向量[J]. 中国新通信,2018,20(23):192 - 193.

[13] 唐琳,何天宇. 基于 Python 的自然语言数据处理系统的设计与实现[J]. 电子技术与软件工程,2018(16):160 - 162.

[14] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. Journal of Machine Learning Research, 2012,13:281 - 350.

[15] 陆高. 基于智能计算的超参数优化及其应用研究[D]. 西安:西安电子科技大学,2018.

[16] 崔佳旭,杨博. 贝叶斯优化方法和应用综述[J]. 软件学报,2018,29(10):3068 - 3090.

[17] Shahriari B, Swersky K, Wang Z, et al. Taking the human out of the loop: a review of bayesian optimization[J]. Proceedings of the IEEE, 2016,104(1):148 - 175.

[18] Xu L,Krzyzak A,Suen C Y. Methods of combining multiple classifiers and their applications to handwriting recognition [J]. IEEE Transactions on System, Man and Cybernetics, 1992,22(3):418 - 435.

[19] 蒋芸,陈娜,明利特,等. 基于 Bagging 的概率神经网络集成分类算法[J]. 计算机科学,2013,40(5):242 - 246.

[20] Zhang L, Suganthan P N. Oblique decision tree ensemble via multisurface proximal support vector machine [J]. IEEE Transactions on Cybernetics,2015,45(10):2165 - 2176.

[21] 范永东. 模型选择中的交叉验证方法综述[D]. 太原:山西大学,2013.


(上接第 151 页)


[13] 冀俊忠,庞皓明,杨翠翠,等. 基于多隐层极限学习机的文本分类方法[J]. 北京工业大学学报,2019,45(6):534 - 545.

[14] 王田田,王艳,纪志成. 基于改进极限学习机的滚动轴承故障诊断[J]. 系统仿真学报,2018,30(11):4413 - 4420.

[15] 崔鹏宇,王泽勇,邱春蓉,等. 基于多尺度排列熵与双核极限学习机的滚动轴承故障诊断方法[J]. 电子测量与仪器学报,2019,33(5):142 - 147.

[16] Wei Z X,Wang Y X,He S L,et al. A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection [J]. Knowledge-Based Systems,2017,116:1 - 12.

[17] Liao S Z,Feng C. Meta-ELM:ELM with ELM hidden nodes [J]. Neurocomputing,2014,128:81 - 87.