文章编号：2095-2163(2019)05-0150-05 中图分类号：TP391.1 文献标志码：A

基于朴素贝叶斯的文本情感分类及实现

梁 柯，李 健，陈颖雪，刘志钢
（上海工程技术大学，上海 201620）

摘 要： 本文利用 Python 语言，对 25 000 条英文影评数据进行文本分类。首先利用词袋模型对文本数据进行分类。在此基础上加入 Word2Vec 建立新的词向量特征，通过精准率和召回率对比前后 2 种模型的分类效果；最后通过逻辑回归和朴素贝叶斯分类模型的分类效果对照得出研究结论。结果表明：对于英文影评文本分类，在同等条件下，使用 Word2Vec 构建词向量模型的精准率和召回率比使用 bag of Word 词袋模型分别高出 0.02 个百分点和 0.026 个百分点；在使用 Word2Vec 的基础上，朴素贝叶斯分类器的精准率和召回率分别高出逻辑回归分类 0.027 个百分点和 0.028 个百分点。

关键词： 文本分类；词袋模型；Word2Vec；逻辑回归；朴素贝叶斯

Text emotional classification and realization based on Naive Bayes

LIANG Ke, LI Jian, CHEN Yingxue, LIU Zhigang
(Shanghai Unviersity of Engineering Science, Shanghai 201620, China)

【Abstract】 This paper uses Python language to categorize 25 000 English film review data. Firstly, word bag model is used to categorize text data; Then, Word2Vec is added to build new feature vectors, and the classification results of the two models are compared by precision and recall rates; Finally, the classification effects of logistic regression and Naive Bayesian classification model are compared. The results show that the accuracy and recall rate of word vector model using Word2Vec are 0.02 and 0.026 percentage points higher than those using bag of Word model under the same conditions. On the basis of Word2Vec, the accuracy rate and recall rate of Naive Bayes classifier are 0.027 and 0.028 percentage points higher than Logistic Regression classification respectively.

【Key words】 text classification; word bag model; Word2Vec; Logistic Regression; Naive Bayes

0 引 言

互联网已融入社会生活的方方面面[1]，各种通讯电子等行业迅速发展，使得文本、图像、视觉等数据挖掘任务的需求不断增加，而文本分类技术现已广泛应用到信息过滤、信息检索、词义消歧、信息组织及管理、话题发现及跟踪等多个。研究可知，文本分类能够帮助用户对庞杂的数据信息进行精准分类，或是帮助用户快速定位和筛选所需信息，从海量文本数据中挖掘出对于当前个人用户或用户群最具价值的信息则有着较高的研究意义和应用价值。

分类是机器学习和数据挖掘领域中一项重要任务。分类是把数据样本映射到一个事先定义的类中的学习过程，其实质就是根据现有的样本组成的训练集判断一个新样本的类别。严格来讲，分类也是一种预测，是对一组离散属性（类标号）的预测，而预测通常指的是对连续值属性的估计[2]分类，现已成为众多领域的关键技术，诸如情感分类[3-4]、自然语言处理[5]、计算机视觉[6]、手写文字[7]等等。常用的分类算法有 k 近邻、决策树、随机深林、逻辑回归、贝叶斯、神经网络等模型[8]。而文本分类任务大多都是二分类任务，最常用的方法是逻辑回归和朴素贝叶斯分类模型。

基于此，本文运用 kaggle 网 5 000 条影评数据，并选取 Python 程序语言，来进行影评数据的情感分类研究。对此，本文拟展开分析论述如下。

1 英文文本预处理

数据预处理是数据挖掘的重要部分，在真实世界中，数据通常是不完整的（缺少某些感兴趣的属性值）、不一致的（包含代码或者名称的差异）、极易受到噪声（错误或异常值）的侵扰的。研究中对这些噪声数据进行处理，不仅有利于后续数据分析，并可使数据分析结果更有意义。

无论是中文还是英文数据，文本数据的预处理一般包括 5 个步骤，即：去掉 html 标签、移除标点、

作者简介：梁 柯（1994-），男，硕士研究生，主要研究方向：轨道交通运营管理（客流分析与预测）；李 健（1980-），男，博士，讲师，主要研究方向：城市轨道交通人因工程、轨道交通政策法规与经济评价；陈颖雪（1983-），女，博士，讲师，主要研究方向：轨道交通客流特征分析与预测；刘志钢（1974-），男，博士，教授，主要研究方向：轨道交通运营管理优化及安全技术、轨道交通人因工程。

收稿日期：2019-07-08

切分成词、去掉停用词和重组为新的句子。这里对各步骤的设计分析可做重点阐述如下。

1.1　去掉 Html 标签

在当今互联网时代，对文本数据来说，来源广泛，易获得，但质量参差不齐。考虑到很多数据都是从互联网上实时爬取得来，而爬取得到的数据会含有大量的 Html 网页标签和表情等，但 Html 标签对数据分析没有任何作用，甚至还会影响分析效果。综上分析可知，从互联网上取得数据后的第一步就是去除 Html 标签。

1.2　去掉标点符号

对于文本分类而言，标点符号和特殊符号的存在影响计算机识别效果，为了确保分类器的分类速度较快，以及得到良好分类的准确率，故需过滤掉这些噪声数据[9]。

1.3　文本分词

无论是在汉语还是英语中，词一般都代表最小的语义单位，因此在研究中就需要将句子划分成词，才可转入后续的研究分析中。在 Python 语言中，中文分词一般选用 Jieba 分词器，英文分词一般选用 Nltk 分词器。其中，Jieba 是一款基于 Python 的中文分词器，目前也是一款流行的开源分词器，内部有多个算法，支持多种分词模式，并可以利用隐马尔可夫模型和维特比算法解决部分未登录词问题。Nltk（Natural Language Toolkit），是自然语言处理工具包，也是 NLP 领域中，最常使用的一个 Python 库，Nltk 包括图形演示和示例数据。

1.4　去掉停用词

去除停用词可以大大减小特征词的数量，进而提高文本分类的准确性。停用词主要有 2 种类型。一种是人类语言中包含的功能词。这些功能词都较为常见，类似虚词，与其它词相比，没什么实际意义。比如英语中的 the、is、at、which、on 等；另一种是词汇词，比如 want 等。对中文来说，包括着诸如“的”、“和”、“在”、“是”等在内的一系列副词、量词、介词、叹词、数词。对文本分类来说，这些词汇几乎在所有文本中都会出现，不具备特殊性和区分度，反而会稀释那些有区分度的词，所以通常会把这些词从问题中移去，如此一来就提高了分类性能。

1.5　数据重组

数据重组是文本分类中的重要环节。数据重组就是将预处理后的文本数据重新组成一句完整的话，便于后面的词向量构建和模型的训练。

2　文本特征抽取和词向量模型

2.1　文本特征抽取

文本特征抽取是为了提高文本分类的效率，减少计算复杂度。研究时，可通过判断特征词来进行文本特征选择。总地来说，文本特征选取方法有文本频率和词频两种，其它包括卡方检验、信息增益、互信息等方法是以文档频率为基础，常用的 TF-IDF 则是综合词频和文档频率构建的特征选择方法。在本文研究中，选用了词频统计方法进行文本分类处理。

2.2　词向量模型

词向量模型是将文本文件表示为标识符（如索引）向量的代数模型。其主要适用于信息过滤、信息检索、索引和相关排序方面的研究。词向量就是用一个向量的形式表示一个词。机器学习任务则是把任何输入量化成数值表示，同时再充分利用计算机的运算能力，计算求出最终想要的结果。

本文分别使用 bag of Word 模型和 Word2Vec 模型进行词向量构建，并且对比分析了 2 种词向量模型进行文本分类的性能优劣。

3　文本分类模型

常用的文本分类算法有逻辑回归算法、朴素贝叶斯算法和神经网络算法。其中，逻辑回归是经典的二分类算法，既可用于预测，也能用于分类。研究选用 Sigmoid 函数，将输入映射为概率值，实现预测功能，通过设置概率阈值实现分类功能。逻辑回归不仅能够得到较好的分类效果，而且算法简单明了，在机器学习分类算法选择中，逻辑回归已然成为二分类任务首选算法，但是在数据特征有缺失或者特征空间很大时的运算效果并不好。研究推得其数学运算公式可表示为：

正例：$P(y=1 \mid x;\varnothing)=h_{\varnothing}(x)$，　　(1)

负例：$P(y=0 \mid x;\varnothing)=1-h_{\varnothing}(x)$，　　(2)

整合：

$$P(y \mid x;\varnothing)=(h_{\varnothing}(x))^{y}(1-h_{\varnothing}(x))^{1-y}, \quad (3)$$

朴素贝叶斯算法是一种基于贝叶斯定理和特征条件独立假设的分类方法，其应用领域较为广泛。贝叶斯分类器所需估计的参数很少，对缺失数据不太敏感，算法也比较简单，可解释性强。理论上，与其它分类方法相比具有最小的误差率。通常，对文本进行分类是为了创建一个属性模型，对于不相互独立的属性，也可单独进行处理。公式如下：

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{P(B)}. \quad (4)$$

神经网络是机器学习中的常用算法,这是一种应用类似于大脑神经突触联接的结构进行信息处理的数学模型,通过模拟大脑神经元来输入数据,运用一定的算法得到运算结果,再根据结果值计算损失值,利用损失值反向调整模型参数,直至使得模型达到最优的过程。神经网络模型的出众计算能力、自学习和自适应能力以及泛化能力强等优点使得深度学习在当今学界已成为研究热点,但神经网络模型也存在着无法探知其中间的学习过程,容易出现过拟合现象,且对于小规模数据集闭合效果欠佳等缺点。常用的 BP 神经网络模型的总体架构包括输入层、隐层和输出层,卷积神经网络则在隐层中加入了卷积层和池化层。BP 神经网络和卷积神经网络的模型设计架构分别如图 1、图 2 所示。

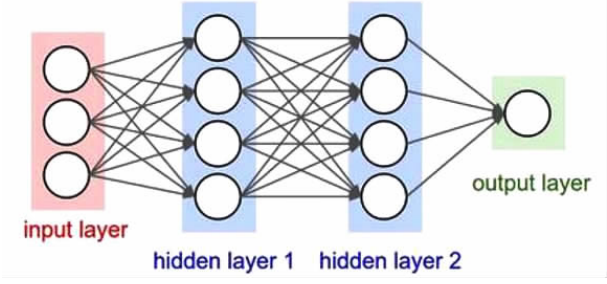

图 1 BP 神经网络模型架构

Fig. 1 BP neural network model architecture

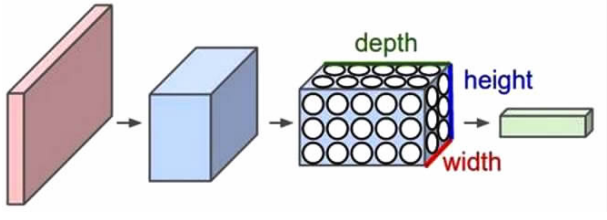

图 2 卷积神经网络模型架构

Fig. 2 Convolutional neural network model architecture

4 基于 Python 的英文文本分类设计实现

本文选取 kaggle 网上 25 000 条英文影评数据进行实例分析,通过考察用户评论,对该评论进行情感分类,分为积极(Positive)和消极(Negative)两类影评数据,并选用精准率和召回率来评价分类效果。基于前述分析流程,逐步实现分类任务,包括词向量构建对比分类和分类器对比分类。部分数据如图 3 所示。对此研究过程,可得阐释分述如下。

```
df = pd.read_csv('../data/labeledTrainData.tsv', sep='\t', escapechar='\\')
print('Number of reviews: {}'.format(len(df)))
df.head()
```

```
Number of reviews: 25000
```

	id	sentiment	review
0	5814_8	1	With all this stuff going down at the moment w...
1	2381_9	1	"The Classic War of the Worlds" by Timothy Hin...
2	7759_3	0	The film starts with a manager (Nicholas Bell)...
3	3630_4	0	It must be assumed that those who praised this...
4	9495_8	1	Superbly trashy and wondrously unpretentious 8...

图 3 影评文本数据

Fig. 3 Film review text data

4.1 数据预处理

根据上文可知,数据预处理包括 5 个部分,即:去掉 Html 标签、去除标点符号、切分成词、去掉停用词和数据重组。程序代码如图 4 所示。将预处理好的数据进行重组,命名为 clean_review,详见图 5。

```
eng_stopwords = set(stopwords)

def clean_text(text):
    text = BeautifulSoup(text, 'html.parser').get_text()#去html标签
    text = re.sub(r'[^a-zA-Z]', ' ', text)#去标点符号
    words = text.lower().split()#分词
    words = [w for w in words if w not in eng_stopwords] #去停用词
    return ' '.join(words)
```

图 4 数据预处理

Fig. 4 Data preprocessing

清洗数据添加到dataframe里

```
df['clean_review'] = df.review.apply(clean_text)
df.head()
```

	id	sentiment	review	clean_review
0	5814_8	1	With all this stuff going down at the moment w...	stuff moment mj ve started listening music wat...
1	2381_9	1	"The Classic War of the Worlds" by Timothy Hin...	classic war worlds timothy hines entertaining ...
2	7759_3	0	The film starts with a manager (Nicholas Bell)...	film starts manager nicholas bell investors ro...
3	3630_4	0	It must be assumed that those who praised this...	assumed praised film filmed opera didn read do...
4	9495_8	1	Superbly trashy and wondrously unpretentious 8...	superbly trashy wondrously unpretentious explo...

图 5 数据重组

Fig. 5 Data reorganization

4.2 特征抽取和词向量模型

本文运用词频方法抽取数据特征(用 sklearn 的 CountVectorizer),分别选用 bag of Word 和 Word2Vec 两种词袋模型构造词向量,部分代码参见图 6。

```
vectorizer = CountVectorizer(max_features = 5000)
train_data_features = vectorizer.fit_transform(df.clean_review).toarray()
train_data_features.shape
```

```
(25000, 5000)
```

图 6 词向量构建模块

Fig. 6 Word vector construction module

4.3 文本分类模型

4.3.1 逻辑回归分类

使用 bag of Word 模型的逻辑回归分类,部分代码如图 7 所示。

```
LR_model = LogisticRegression()
LR_model = LR_model.fit(X_train, y_train)
y_pred = LR_model.predict(X_test)
cnf_matrix = confusion_matrix(y_test,y_pred)

print("Recall metric in the testing dataset: ", cnf_matrix[1,1]/(cnf_matrix[1,0]+cnf

print("accuracy metric in the testing dataset: ", (cnf_matrix[1,1]+cnf_matrix[0,0])/
```

```
Recall metric in the testing dataset:  0.853181076672
accuracy metric in the testing dataset:  0.8454
```

图 7　使用 bag of Word 逻辑回归分类结果

Fig. 7　Logistic regression classification results using bag of Word

使用 bag of Word 预料模型的逻辑回归分类效果见图 7，准确率为 0.845，召回率为 0.853，分类效果较好，其混淆矩阵如图 8 所示。

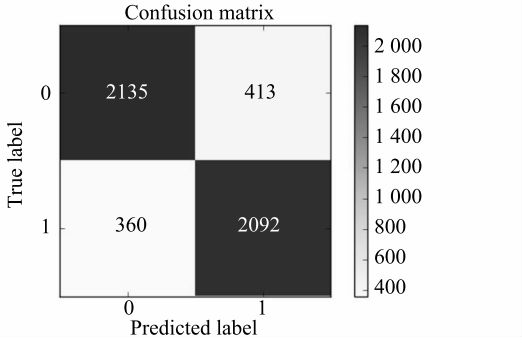

图 8　混淆矩阵

Fig. 8　Confusion matrix

使用 Word2Vec 模型的逻辑回归分类中，词向量维度设为 300，最小单词数量设为 40，单词移动窗口设为 10。分类结果如图 9 所示。

```
LR_model = LogisticRegression()
LR_model = LR_model.fit(X_train, y_train)
y_pred = LR_model.predict(X_test)
cnf_matrix = confusion_matrix(y_test,y_pred)

print("Recall metric in the testing dataset: ", cnf_matrix[1,1]/(cnf_matrix[1,0]+cnf_matrix[1,1]))

print("accuracy metric in the testing dataset: ", (cnf_matrix[1,1]+cnf_matrix[0,0])/(cnf_matrix[0,0]+cnf_matrix[
```

```
Recall metric in the testing dataset:  0.87969004894
accuracy metric in the testing dataset:  0.865
```

图 9　使用 Word2Vec 的逻辑回归分类结果

Fig. 9　Results of logistic regression classification using Word2Vec

由图 9 可知，使用 Word2Vec 模型的逻辑回归比使用 bag of Word 的精准率提高了 0.02 个百分点，召回率提高了 0.026 个百分点。

4.3.2　朴素贝叶斯分类

通过以上分析，使用 Word2Vec 构建词向量的逻辑回归分类效果更好。本节在 Wor2Vec 的基础上，选用朴素贝叶斯分类器，分别采用逻辑回归和朴素贝叶斯算法对影评数据的分类效果进行比较。研究最终得到的朴素贝叶斯的分类结果如图 10 所示。

由图 10 可知，朴素贝叶斯分类器比逻辑回归分类精准率高出了 0.027 个百分点，召回率高出了 0.028个百分点。由此可见，在影评文本分类中，朴素贝叶斯分类器的分类效果要优于逻辑回归。

```
N_model =MultinomialNB(alpha=1.0)
N_model = N_model.fit(X_train, y_train)
y_pred = N_model.predict(X_test)
cnf_matrix = confusion_matrix(y_test,y_pred)

print("Recall metric in the testing dataset: ", cnf_matrix[1,1]/(cnf_matrix[1,0]+cnf_matrix[1,1]))

print("accuracy metric in the testing dataset: ", (cnf_matrix[1,1]+cnf_matrix[0,0])/(cnf_matrix[0,0]+cnf_matrix[1,
```

```
Recall metric in the testing dataset:  0.8974714518760196
accuracy metric in the testing dataset:  0.892
```

图 10　朴素贝叶斯分类结果

Fig. 10　Classification results of naive bayes

5　结束语

本文讨论了文本分类常用的分类方法，即逻辑回归和朴素贝叶斯算法。与此同时又可看到，近年来已陆续涌现数目可观的利用深度学习进行文本分类的研究，并在许多公开数据集和分类任务上都取得了最优结果。但仍要指出，深度学习也有着严重缺陷，对此可做详尽剖述如下。

（1）需要大量的数据。深度学习是一种数据驱动型的技术，海量的数据与深度学习算法结合往往能带来巨大的效果提升，但如果数据量不足（比如本文中用到的新闻分类数据集）时，深度学习算法容易出现过拟合，泛化效果很差。

（2）缺乏解释性。深度学习端到端的训练和学习带来很多便捷，无须人工繁杂地提取特征，也无须设计过多中间步骤，但这种端到端也会带来不可预测的黑箱效应。参数调节与最终结果的好坏难以做到一一对应，缺乏指导意义，并且很难复现模型。鉴于前述分析可知，朴素贝叶斯在现今工业界也仍会受到青睐。

本文利用 Python 语言，对 25 000 条英文影评数据进行文本分类。首先利用词袋模型对文本数据进行分类；在此基础上加入 Word2Vec 建立新的词向量特征，通过精准率和召回率对比前后两种模型的分类效果；最后，即对逻辑回归和朴素贝叶斯分类进行了对比研究。结果表明，对于英文影评文本分类，在同等条件下，使用 Word2Vec 构建词向量模型优于使用 bag of Word 词袋模型，朴素贝叶斯分类效果优于逻辑回归分类。在后续工作中，尤其在诸如决策树、神经网络等算法对比上仍有待进一步的深入探索与研究。

参考文献

[1] Miniwatts Marketing Group. 世界互联网统计网站[EB/OL]. [2019-08-21]. http://www.internetworldstats.com.

（下转第 157 页）