基于NLTK的电影评论的情感分析

|  |  |  |
| --- | --- | --- |
| 姓名 | 专业 | 学号 |
| 邓江 | 计算机应用技术 | 231017000149 |

一 研究背景

本文利用电影评论，并对评论中所表达的情感进行分析和识别。对电影评论的研究，可以帮助观众了解电影的质量、风格和内容。 电影评论对观众的观影行为，票房收入、口碑传播、影片制作与发行等都有极大的影响。基于上述问题，提出基于电影评论文本分类系统的实现。本文采用朴素贝叶斯分类器进行训练和评估。主旨是通过模型构建，更好的理解自然语言处理（NLP）。

二 研究目标及内容

基于NLTK构建模型, 对文本数据集进行训练然后对测试文本预测其分类。

三 研究步骤

1. 准备数据集
   1. 数据集说明

由于NLTK官网提供了关于电影评论的训练语料库，本项目采用NLTK提供的movie\_reviews数据集来训练我们的网络模型。本文项目使用的到语料库(corpora)如下：

|  |  |
| --- | --- |
| 文件信息 | 文件大小 |
| movie\_reviews | 12M |
| omw-1.4 | 26M |
| stopwords | 156K |
| wordnet | 11M |

表1.1

上述文件通过如下方式下载：

|  |
| --- |
| 在python中使用如下进行数据集下载：  import nltk  nltk.download('movie\_reviews')  nltk.download('punkt')  nltk.download('omw-1.4')  nltk.download('stopwords')  nltk.download('wordnet') |

* 1. 数据集使用

在代码中设置本地语料库路径，在程序调用中会使用本地指定路径下面的语料库数据。

|  |
| --- |
| import os  import nltk  path = "/root"  os.chdir(path)  from nltk.corpus import movie\_reviews  from nltk.tokenize import word\_tokenize  from nltk.corpus import stopwords  from nltk.stem import WordNetLemmatizer  from nltk.corpus import movie\_reviews  from nltk.classify import NaiveBayesClassifier  from nltk.classify.util import accuracy  from sklearn.feature\_extraction.text import TfidfVectorizer  from nltk.probability import FreqDist |

1. 数据处理

2.1 对数据进行分词：

|  |
| --- |
| # 分词  def tokenize(text):  tokens = word\_tokenize(text)  return tokens |

2.2 去除停用词：

|  |
| --- |
| # 去除停用词  def remove\_stopwords(tokens):  stop\_words = set(stopwords.words('english'))  filtered\_tokens = [token for token in tokens if token.lower() not in stop\_words]  return filtered\_tokens |

2.3词形还原：

|  |
| --- |
| # 词形还原  def lemmatize(tokens):  lemmatizer = WordNetLemmatizer()  lemmatized\_tokens = [lemmatizer.lemmatize(token) for token in tokens]  return lemmatized\_tokens |

2.4 数据集预处理：

|  |
| --- |
| # 对整个数据集进行预处理  documents = [(list(movie\_reviews.words(fileid)), category)  for category in movie\_reviews.categories()  for fileid in movie\_reviews.fileids(category)] |

2.5 数据集训练和评估：

本项目使用朴素贝叶斯分类器进行训练和评估：

|  |
| --- |
| #使用朴素贝叶斯分类器进行训练和评估：  from sklearn.feature\_extraction.text import CountVectorizer  from sklearn.naive\_bayes import MultinomialNB  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import accuracy\_score  from sklearn.utils import shuffle  # 将数据集划分为训练集和测试集  X, y = zip(\*documents)  num\_train\_sets = 20  for i in range(num\_train\_sets):  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25)  X\_train = [str(x) for x in X\_train]  X\_test = [str(x) for x in X\_test]  # 将文本转换为特征向量  vectorizer = TfidfVectorizer()  #vectorizer = CountVectorizer()  X\_train\_vec = vectorizer.fit\_transform(X\_train)  X\_test\_vec = vectorizer.transform(X\_test)  # 使用朴素贝叶斯分类器进行训练  clf = MultinomialNB()  clf.fit(X\_train\_vec, y\_train)  # 预测测试集  y\_pred = clf.predict(X\_test\_vec)  # 计算准确率  accuracy = accuracy\_score(y\_test, y\_pred)  print("Accuracy: ", accuracy) |

2.6 评估结果：

|  |
| --- |
| [root@iZbp1enokt3ghl7pc7peo7Z ~]# python3.6 231017000149\_nltk.py  Accuracy: 0.818  Accuracy: 0.83  Accuracy: 0.768  Accuracy: 0.792  Accuracy: 0.8  Accuracy: 0.77  Accuracy: 0.806  Accuracy: 0.796  Accuracy: 0.84  Accuracy: 0.786  Accuracy: 0.814  Accuracy: 0.74  Accuracy: 0.812  Accuracy: 0.786  Accuracy: 0.792  Accuracy: 0.834  Accuracy: 0.774  Accuracy: 0.796  Accuracy: 0.79  Accuracy: 0.76 |

1. 项目结果

3.1 使用模型预测

|  |
| --- |
| #新评论数据测试  X\_test = ["this is a very bad films", "This movie was truly a cinematic masterpiece. From the plot to the character development, and even the visual effects, every aspect was filled with innovation and surprises. I particularly admired how the director skillfully incorporated complex emotional threads into the story, making the audience both nervous and excited while watching. The performances of the actors were also outstanding, as they successfully portrayed unique and relatable characters. The musical score further enhanced the film, adding more drama to the plot's progression. Overall, it is a well-crafted and captivating movie that is definitely worth watching again and again."]  # 将测试数据转换为特征向量  X\_test\_vec = vectorizer.transform(X\_test)  # 使用模型进行预测  print('对输入的新评论进行预测：')  predictions = clf.predict(X\_test\_vec)  print(predictions) |

3.2预测结果：

|  |
| --- |
| 对输入的新评论进行预测：  ['neg' 'pos'] |

3.3结果分析

本项目采用NLTK提供的语料库对电影评论的情感分类的例子，根据不通的应用场景和角度对电影进行适当的调整，以满足各方不同的需求。该模型中，通过调整训练模型的参数，通过多次训练后，改善模型分析的准确性，以便提供更加准确的情感分类。

四 总结

通过本次项目实验，对上课讲解的知识有了更深一步的了解，对NLP有了初步的认知，希望在后续的工作中，可以结合目前了解和学习的知识，运用到工作实践当中。在此感谢老师的辛苦付出和耐心指导。

五 参考文献

1. <https://www.nltk.org/>
2. <https://blog.csdn.net/u010349629/article/details/130663037>