



東華理工大學

EAST CHINA UNIVERSITY OF TECHNOLOGY

毕业论文

题目 基于机器学习的新闻评论情感分析方法研究

英文题目 Research on Machine Learning-based Sentiment

Analysis Methods for News Commentary

学生姓名: XXX 申请学位门类: 工学

学 号: XXXXXXXX

专 业: XXXXXX

学 院: XXXXXX

指导教师: XXXXX 职称: XXXXX

二〇二四年五月一日

作 者 声 明

本人以信誉郑重声明：所呈交的学位毕业设计（论文），是本人在指导教师指导下由本人独立撰写完成的，没有剽窃、抄袭、造假等违反道德、学术规范和其他侵权行为。文中引用他人的文献、数据、图件、资料均已明确标注出，不包含他人成果及为获得东华理工大学或其他教育机构的学位或证书而使用过的材料。对本设计（论文）的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本毕业设计（论文）引起的法律结果完全由本人承担。

本毕业设计（论文）成果归东华理工大学所有。
特此声明。

毕业设计（论文）作者（签字）：

签字日期： 年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

随着互联网的发展,新闻作为一种重要的信息传播媒介,不仅影响着公众的认知和观点,也反映了社会的热点和趋势。新闻评论的数量庞大,内容复杂,语言多样,难以用传统的方法进行有效地处理和分析。因此,利用人工智能技术,特别是文本情感分析技术,对新闻评论进行情感倾向的识别和分类,是一项具有挑战性和意义的研究课题。

本项目基于 Python 语言实现,对凤凰新闻网的新闻文章用户评论数据进行情感分析。首先,利用 Urllib 爬虫采集新闻评论数据。其次,运用 Pandas 库对采集后的原始数据进行预处理,包括运用 jieba 库进行中文文本分词、去除停用词、使用正则表达式对文本进行清洗。然后,分别利用词袋模型、TF-IDF 算法、Word2vec 三种特征工程分析方法对预处理后的新闻文本数据进行特征抽取,将非结构化的自然语言文本转换为结构化的、计算机可以理解和处理的数学特征向量。接着,分别使用决策树、朴素贝叶斯、支持向量机三种机器学习模型对新闻评论文本进行情感分类。采用精确率、召回率和 F1 值三个指标对三种算法得到的情感分析结果进行对比评价。最后,基于以上新闻评论数据情感分析模型,本文构建了一个新闻评论情感分析交互系统。系统可以根据用户输入的凤凰新闻网页 URL,自动获取该新闻的所有用户评论,并调用保存好的模型进行新闻评论情感分析,生成“每日评论数量变化”折线图和用户评论词云。

关键词: 新闻评论; 情感分析; 机器学习

ABSTRACT

With the development of the Internet, news, as an important medium of information dissemination, not only influences public perceptions and opinions, but also reflects the hotspots and trends in society. The large number of news commentaries, the complexity of their contents, and the diversity of their languages make it difficult to be processed and analysed effectively by traditional methods. Therefore, it is a challenging and meaningful research topic to identify and classify the emotional tendency of news commentaries by using artificial intelligence technology, especially text sentiment analysis technology.

This project is based on the implementation of Python language to perform sentiment analysis on the news article user comment data of Phoenix News. Firstly, Urllib crawler is used to collect news comment data. Secondly, the Pandas library is used to pre-process the raw data after collection, including the use of jieba library for Chinese text segmentation, removal of deactivated words, and the use of regular expressions to clean the text. Then, three feature engineering analysis methods, namely bag-of-words model, TF-IDF algorithm, and Word2vec, were used to extract features from the preprocessed news text data respectively, and to convert unstructured natural language text into structured mathematical feature vectors that can be understood and processed by computers. Then, three machine learning models, namely, decision tree, plain Bayes, and support vector machine, were used to classify the news commentary text for sentiment classification, respectively. The three indicators of precision rate, recall rate and F1 value are used to compare and evaluate the sentiment analysis results obtained by the three algorithms. Finally, based on the above news review data sentiment analysis model, this paper constructs a news review sentiment analysis interactive system. The system can automatically obtain all the user comments of the news according to the URL of the Phoenix News webpage entered by the user, and call the saved model to perform the sentiment analysis of the news comments, generating a line graph of the ‘daily change in the number of comments’ and a word cloud of user comments.

Keywords: news commentaries; sentiment analysis; machine learning

目 录

摘 要.....	I
ABSTRACT.....	II
第 1 章 绪论.....	1
1.1 选题背景与意义.....	1
1.2 情感分析相关技术及理论.....	1
1.3 技术路线.....	2
1.4 本论文的主要结构.....	3
第 2 章 新闻评论数据获取与预处理.....	4
2.1 新闻评论数据的采集.....	4
2.2 新闻评论数据的预处理.....	4
2.2.1 文本分词.....	5
2.2.2 去除停用词.....	5
2.2.3 正则清洗.....	6
第 3 章 新闻评论数据特征选择.....	7
3.1 词袋模型 (Bag-of-Words model).....	7
3.2 词频-逆文档频率 (TF-IDF).....	10
3.3 Word2Vec.....	11
第 4 章 新闻评论情感分析算法实现.....	13
4.1 数据集划分.....	13
4.2 基于决策树分类算法的新闻评论情感分析.....	13
4.2.1 训练模型.....	13
4.2.2 模型测试.....	14
4.2.3 词袋模型的决策树模型调优及保存.....	16
4.2.4 TF-IDF 算法的决策树模型调优及保存.....	17
4.3 基于朴素贝叶斯分类算法的新闻评论情感分析.....	19
4.3.1 训练模型.....	19
4.3.2 模型测试.....	21
4.3.3 TF-IDF 算法的朴素贝叶斯模型调优及保存.....	21
4.4 基于支持向量机分类算法的新闻评论情感分析.....	22
4.4.1 训练模型.....	22

4.4.2 模型测试	23
4.4.3 TF-IDF 算法的支持向量机模型调优及保存.....	23
4.4.4 Word2vec 算法的支持向量机模型调优及保存.....	25
4.5 新闻评论情感分类结果分析与总结.....	26
第 5 章 新闻评论情感分析系统实现.....	30
5.1 爬虫模块.....	30
5.2 词云模块.....	31
5.3 折线图模块.....	31
5.4 前端页面.....	32
结束语.....	35
致 谢.....	36
参考文献.....	37

第1章 绪论

1.1 选题背景与意义

随着互联网的发展,新闻作为一种重要的信息传播媒介,不仅影响着公众的认知和观点,也反映了社会的热点和趋势。新闻评论作为新闻的补充和延伸,更能体现出网民的情感和态度,对于分析社会舆情和公共情绪具有重要的价值。然而,新闻评论的数量庞大,内容复杂,语言多样,难以用传统的方法进行有效地处理和分析。因此,利用人工智能技术,特别是文本情感分析技术,对新闻评论进行情感倾向的识别和分类,是一项具有挑战性和意义的研究课题。

本论文的研究和实现,对于以下几个方面具有重要的意义:

(1) 对于新闻媒体和编辑,能够及时了解新闻的受欢迎程度和网民的反馈,调整新闻的选题和报道策略,提高新闻的影响力和传播效果。

(2) 对于政府和管理部门,能够掌握社会的热点和舆情,发现和解决社会问题,维护社会稳定和谐。

(3) 对于学术界和研究者,能够探索和应用新的文本情感分析方法和技术,提高文本情感分析的准确性和效率,丰富文本情感分析的应用领域和场景。

(4) 对于普通用户,能够方便地浏览和查看新闻评论的情感分析结果,增加新闻阅读的趣味性和互动性,拓宽新闻阅读的视野和角度。

因此,利用人工智能技术进行全自动剖析这种主观文本中所体现的感情变成了大数据挖掘科学研究的一个热点。

本文获取凤凰新闻网站中任意文本新闻的相关用户评论,并使用不同的情感分析方法来分析用户评论中的情感态度,并进行对比分析,观察讨论不同情感分析方法的结果准确程度,并针对当前得到的结果进行总结和思考。

1.2 情感分析相关技术及理论

中文文本情感分析是一项利用自然语言处理、机器学习和深度学习等先进技术,对中文文本中包含的主观情绪进行识别、提取、量化和总结的技术。它的核心目标是判断和解析文本所表达的情感倾向,包括积极(正面)、消极(负面)或中立态度,并进一步揭示文本中所蕴含的情绪强度、属性及具体的感情色彩。目前,中文情感分析技术主要包括三种方法:

(1) 基于情感词典的文本情感分析方法,此方法依赖于情感词典中提供的带有情感色彩的词汇及其属性,以实现文本情感倾向的分类。其操作流程通常为:输入文本后进行数据预处理,去除无效字符;随后进行分词处理,将句子分解成词语的组合;接着将情感词典中的词语与分词后的文本进行匹配,将情感词典中的词语引入模型进行训练。最终根据设定的情感判断规则输出情感类型。

(2) 基于机器学习的文本情感分析方法，这种方法通过使用统计机器学习算法对大量已标注或未标注的语料进行特征提取和模型训练，进而输出文本的情感分析结果。基于机器学习的情感分类方法主要分为有监督学习、半监督学习和无监督学习三种形式。

(3) 基于深度学习的文本情感分析方法，起源于 2003 年 Bengio 等人提出的神经网络语言模型。该模型采用三层前馈神经网络结构，包括输入层、隐藏层和输出层。输入层的每个神经元代表一个特征，隐藏层的层数和神经元数量由人工设定，而输出层则代表了分类标签的数量。该模型的主要特点是依据上下文信息预测下一个词的内容，而不依赖于人工标注，有效解决了其他文本情感分析方法在处理上下文语义方面的不足。

1.3 技术路线

项目研究分为两个部分，分别是新闻评论情感分类算法实现和整体系统实现。算法实现的目的是训练机器学习新闻文本情感分类模型，评估新闻评论的情感倾向。系统实现的目的是制作用户交互界面，应用已经保存好的新闻评论情感分析模型。其中，算法研究是项目的核心内容。

算法研究部分，本论文选择基于传统机器学习的研究方式。因为基于情感词典的研究方式准确率低，研究流程复杂。而基于深度学习的研究方式需要大量训练数据，相关理论知识学习周期长，故本论文没有使用。

综合考虑了模型准确率，研究流程复杂程度，数据集需求大小，理论知识学习周期等多个方面因素，最终决定选择基于传统机器学习的研究方向进行本论文的算法研究。本论文分别利用基于 Sklearn 库中词袋模型、TF-IDF 算法、Word2vec 三种特征工程分析方法进行文本特征抽取。使用基于 Python 中的 Sklearn 库中的决策树、朴素贝叶斯、支持向量机三种传统机器学习算法进行文本情感分类。通过搭配不同的特征工程和分类算法进行新闻评论情感分析，计算不同特征工程和分类算法的准确度，讨论不同方法的优劣。

算法研究流程图如图 1-1 所示，分为数据集导入、数据预处理、数据特征提取、算法模型训练、模型测试、模型调优和模型保存七个阶段。

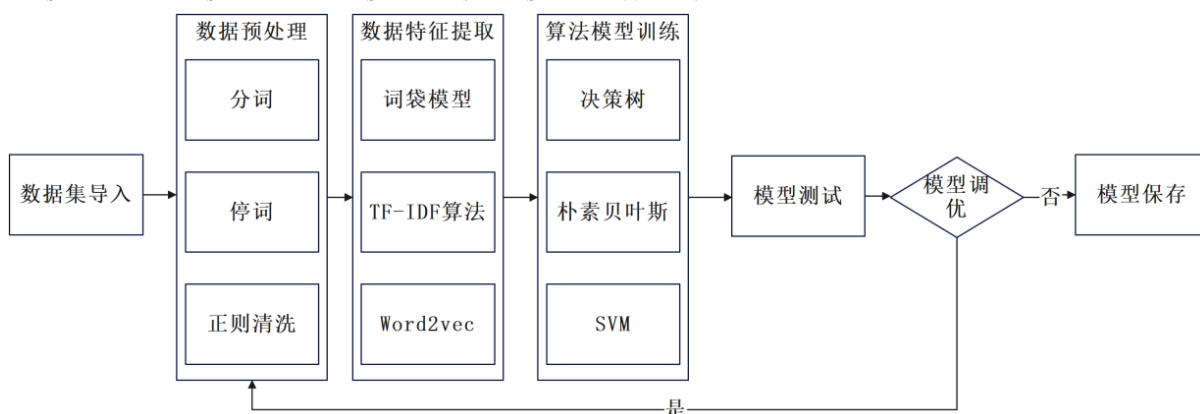


图 1-1 算法研究流程图

系统实现部分，本论文选择的 Tkinter 技术制作用户交互页面，使用 os 技术临时存储数据，使用 Request 技术爬取评论数据。Tkinter 技术可以快速制作界面，所写即所得，方便实时调整。使用 os 技术临时存储系统相关文本数据和图片数据快速方便。Request 技术快捷方便有效，是中小型网络爬虫的首选技术。基于技术实现难度，制作时间等多方面因素考虑，最终选择了上述技术进行本论文的系统实现。本系统可以利用爬虫程序，根据用户输入的凤凰新闻 URL 自动获取该新闻的用户评论数据，调用已经保存到好的情感分析模型和不同特征工程的向量器，自动对每一条用户评论进行情感分析，并输出评论词云和“每日评论数量变化”折线图。

1.4 本论文的主要结构

根据上述方案，可以设计出本论文的主要结构，分为以下几个章节，具体如下：

第 1 章：绪论。阐述这次选题的背景和意义，然后介绍一些情感分析相关技术及理论和研究现状，最后勾勒出本论文主要研究的内容要求、内容以及组织架构。

第 2 章：新闻评论数据的获取与预处理。这次论文机器学习的数据是使用新浪公司在 Github 上开源的旗下新浪微博公司匿名用户评论文本并已经对每条评论进行了情感标注。该数据集非常杂乱并不能直接使用，为了文本之后构建情感分析模型，接下来必须减少一些不适合数据的干扰，在训练模型前进行文本预处理，进行数据清洗，具体分为以下三个处理：去除对情感分析影响较小的词句（去除停用词）、使用正则对文本进行清洗保证只保留中文数据。

第 3 章：新闻评论数据特征选择。这是整个论文的第一个研究核心，这一整个章节是根据情感分析需要，对本文中运用到的文本特征抽取方法的介绍和应用。特征工程是机器学习项目非常重要的一环，直接影响最后分类准确率，

第 4 章：新闻评论情感分析算法实现。这是整个论文的第二个研究核心，这一整个章节是根据之前的实验设计方案和已经完成的特征工程来实现新闻评论情感分类。先是使用不同的特征工程和分类算法进行情感倾向性分类，随后进行实验结果分析总结。

第 5 章：新闻评论情感分析系统实现。构建一个基于 Tkinter 的用户交互界面，集成爬虫模块、词云模块和折线图模块，使得用户只需输入新闻链接，系统就能自动爬取评论数据，调用已训练好的情感分析模型进行情感分类，同时生成反映评论数量变化的折线图和用户评论词云图。

结束语：最后对全论文做一个总结讨论，讨论当前的设计初衷，讨论当前的研究分析总结，反思当前设计研究中所存在的不足，然后对日后应该再如何去完善，未来应该如何去操作进行一个展望。

第2章 新闻评论数据获取与预处理

2.1 新闻评论数据的采集

要对凤凰新闻下的网页新闻的用户评论数据进行分析，就必须先收集评论数据。由于在目前已有的技术中，获取大量新闻评论数据并且人工标注情感特征（积极，消极）时间上和技术上不允许，所以直接使用现有数据集。本项目使用新浪公司在 Github 上开源的其旗下新浪微博公司匿名用户评论文本数据集“weibo_senti_100k.csv”并已经对每条评论进行了情感标注（积极、消极）。数据集共 119989 条用户评论数据，其中 59994 条情感标注为“积极”的用户评论，和 59995 条情感标注为“消极”的用户评论。

数据集链接：[ChineseNlpCorpus/datasets/weibo_senti_100k](https://github.com/SophonPlus/ChineseNlpCorpus/tree/master/weibo_senti_100k) at master · SophonPlus/ChineseNlpCorpus (github.com)。数据集如下图 2-1 所示：



图 2-1 “weibo_senti_100k.csv”数据集

2.2 新闻评论数据的预处理

中文文本预处理是自然语言处理（NLP）领域中的一个重要步骤，它旨在将原始数据集中的中文文本数据转换成适合于后续分析和机器学习模型处理的形式。由于中文文本的特性，如缺少自然分隔符（如英文中的空格）、存在大量的网络用语、表情符号、繁体简体混用等问题，预处理过程显得尤为关键。

由于新浪微博评论原始数据集中含有非常多的语气词，标点符号和表情，这些并不能被机器识别，因此原始数据集并不能被直接使用。为了之后构建情感分析模型，接下来必须减少一些不适合数据的干扰，在训练模型前进行文本预处理，具体分为以下三个

处理：先进行文本分词，之后去除停用词、最后使用正则表达式对文本进行清洗确保只保留中文文本数据。

2.2.1 文本分词

分词是中文文本信息处理中的一个重要阶段，即把文本的每一个有顺序的词句切分成一个个的基本词的过程。这一个阶段，不管是需要情感分类还是要构建词云图，还是计算特征词，都是必不可少的。本文使用基于 Python 的中文文本分词包“jieba”（结巴分词），对原始评论数据集进行中文分词。它的分词原理是利用一个中文词库，确定汉字之间的关联概率，将汉字组成词组，形成分词结果。“结巴分词”实现分词、词性标记、未注册词标识，实现支持用户词典等功用。为更好完成词频统计，分词步骤将词性标注效果去除。

2.2.2 去除停用词

在处理文本数据时，去除停用词是一个重要的预处理步骤。停用词通常是指那些在文本中频繁出现但对于理解文本意义贡献不大的词，例如中文里的“的”、“是”、“啊”等，英文中的“the”、“an”、“their”等。去除这些词可以帮助减少数据的噪声，提高后续处理步骤的效率和准确性。本项目使用的是哈尔滨工业大学在 github 上开源的停用词表，停用词表目前一共收录了 5787 个词，包括中文中对情感分析影响因素较小的词语和标点符号。停用词表链接：[Stopwords/stopwords_hit.txt at main · CharyHong/Stopwords \(github.com\)](https://github.com/CharyHong/Stopwords/blob/master/stopwords_hit.txt)。停用词表如下图 2-2 所示：

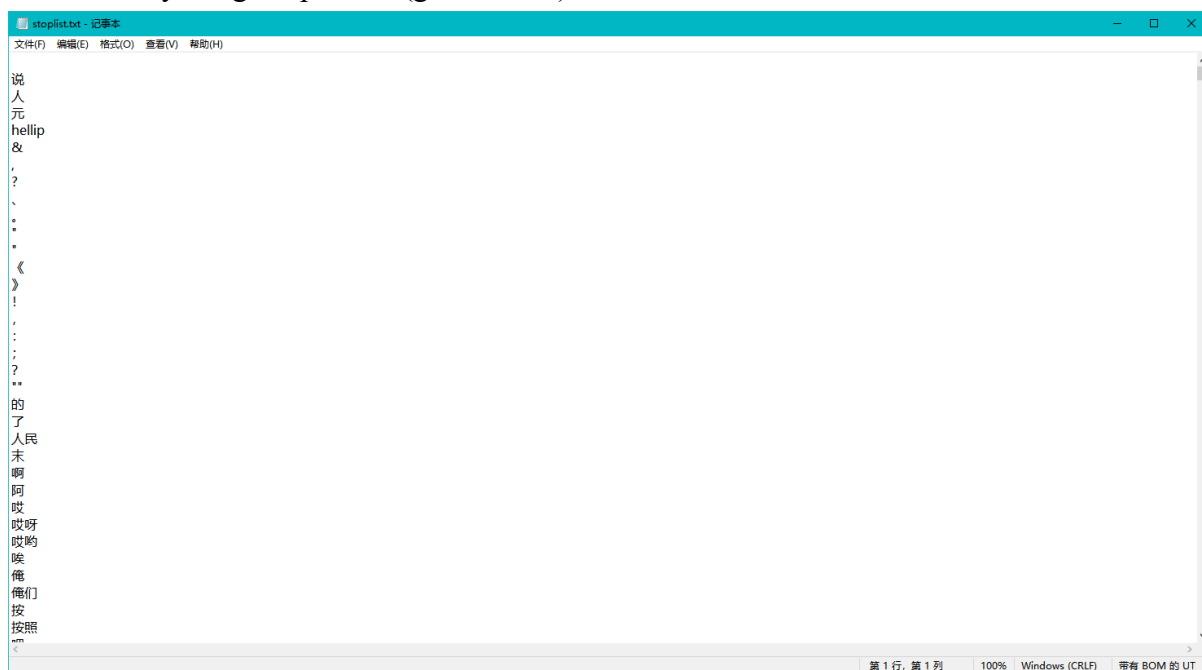


图 2-2 “stoplist.txt” 停用词表

2.2.3 正则清洗

在机器学习项目中，使用正则表达式进行文本清洗是一个常见的预处理步骤。正则表达式是一种强大的文本模式匹配工具，它可以帮助识别和处理复杂的文本模式。在评论数据的预处理阶段，经过分词和去除停用词之后，已经去除了原始数据中的标点符号和停用词，但是原始数据中还有很多特殊符号，表情，阿拉伯数字。这时使用正则表达式来去除这类不必要的字符，以便更好地分析文本内容。

关键代码如下：

```
cleaned_text = re.sub(r'[^\u4e00-\u9fa5]', "", input_text)
```

代码执行主要分为以下四个步骤：

（1）正则表达式模式：`r'[^\u4e00-\u9fa5]'` 定义了一个模式，这个模式匹配所有非中文字符。其中 `\u4e00-\u9fa5` 是中文字符在 Unicode 编码中的范围。

（2）`re.sub` 函数：`re.sub` 函数是 Python 中的一个正则表达式函数，用于替换字符串中与正则表达式匹配的部分。它接受三个参数：模式、替换的字符串（在这里是空字符串）、原始文本。

（3）执行清洗：这个函数将会查找 `input_text` 中所有不在 `\u4e00-\u9fa5` 范围内的字符，并将它们替换为空字符串，也就是说，它会删除所有特殊符号，表情，阿拉伯数字。

（4）输出结果：最后，`cleaned_text` 将包含清洗后的文本，这时，已经可以确保数据 `dataframe` 中只保留了中文文本字符。

第3章 新闻评论数据特征选择

新闻评论数据特征选择，也称为特征提取、特征工程，在自然语言处理（NLP）和文本挖掘领域中，是一个核心步骤，其目的是从原始的文本数据中挑选出最相关、最有信息量的特征（变量或属性），用于后续的分析、建模或预测任务。这个过程不仅能够减少数据的维度，降低计算复杂度，还能剔除噪声，提高模型的性能和泛化能力。

本论文分别使用基于 Sklearn 库中词袋模型（3.1）、TF-IDF 算法（3.2）、Word2vec（3.3）三种特征分析方法进行新闻评论文本特征抽取。

3.1 词袋模型（Bag-of-Words model）

词袋模型（Bag-of-Words, BoW）是自然语言处理（NLP）和信息检索（IR）中的一种基础文本表示技术。这种模型通过简化文本内容，将复杂的文本数据转换为易于处理的数值形式。在词袋模型中，文本的结构和语义复杂性被大幅度降低，因为它不考虑语法和句法规则，也不考虑词语在文本中的顺序和上下文关系。相反，它仅仅关注文本中词汇的存在与否以及它们的出现频次。具体来说，词袋模型将每个文本视为一个装满词语的“袋子”。这些词语可以是单个的单词，也可以是词组（如二元组或三元组，即 n -grams）。在这个模型中，每个词语都被当作一个特征，而文本则被表示为一个向量。在这个向量中，每个维度对应一个词语，其数值代表该词语在文本中的出现次数或者经过特定计算得到的权重。

然而，词袋模型也有其局限性。由于它忽略了词语的顺序，因此无法捕捉到词语之间的语义关系，这可能会导致信息的丢失。例如，“不是很好”和“很不好”这两个短语在词袋模型中可能被视为相同。此外，这种模型也不能处理同义词和多义词的问题，因为它无法识别一个词语在不同上下文中可能有不同的含义。

以下是词袋模型的关键特性及构建过程：

词汇表（Vocabulary）：首先，从整个文本集合中创建一个唯一的词汇表，包含所有可能出现的单词。词汇表中的每个单词都有一个唯一的索引，用于后续构建特征向量。

关键代码：

```
temp_data_list = []
for word, flag in words_pos:
    temp_data_list.append({
        '分词索引': index + 1,
        '词语': word,
        '词性': flag,
        '标签': row['label']
    })
```

创建词汇表以便下一步词袋模型的生成，词典创建如下图 3-1 所示：

11 行 1791564 rows × 4 columns				
分词索引	词语	词性	标签	
0	1 更博	v	1	
1	1 爆照	v	1	
2	1 帅	a	1	
3	1 越来越	d	1	
4	1 爱	v	1	
...
1791559	119988 也	nr	0	
1791560	119988 鬼	n	0	
1791561	119988 知入	v	0	
1791562	119988 睇	vg	0	
1791563	119988 睇	vg	0	

图 3-1 词袋模型词汇表

每个文本（文档）被表示为一个固定长度的向量，向量的维度等于词汇表的大小。向量的每个元素对应词汇表中一个单词的索引位置，元素值通常表示该单词在该文本中的统计量。向量中的值反映了文本中各个单词的相对重要性。

创建词转向量，one-hot 编码关键代码：

```
count_vec = CountVectorizer(binary=True, decode_error="replace")
```

```
x_train = count_vec.fit_transform(x_train)
```

```
x_test = count_vec.transform(x_test)
```

首先，创建了一个 `CountVectorizer` 对象并赋值给变量 `count_vec`。

`CountVectorizer` 用于将文本数据转换为词频向量表示，即将每个文档表示为一组词项（token）的计数构成的向量。这里指定了两个参数：`binary=False`：它意味着输出的向量将表示词汇表中每个单词在文档中出现的次数，而不仅仅是它是否出现过。这种方法称为词频统计（Term Frequency），它考虑了单词的出现频率，而不只是出现的事实。这种设置对于那些单词的出现频率对于文档的意义很重要的情况特别有用，例如在主题建模或文档分类任务中。它可以帮助模型更好地理解文档的内容和上下文。

其次，使用 `count_vec` 对象对训练集 `x_train` 进行词频向量转换。转换后的训练集表示为一个稀疏矩阵（通常是 `scipy.sparse.csr_matrix` 类型），并赋值给 `x_train`。

最后，使用已经生成的词汇表和参数的 `count_vec` 对象对测试集 `x_test` 进行词频向量转换。由于测试集与训练集应遵循相同的预处理规则，这里只需调用 `transform()` 方法。同样，`x_test` 会被转换为与 `x_train` 具有相同词汇表和特征表示形式的稀疏矩阵。

得到的 `x_train` 和 `x_test` 作为预处理后的特征矩阵，可以作为输入提供给后续的机器学习模型进行训练和预测。在这种情况下，特征值为任意数（当 `binary=False` 时），表示对应词项在文档中是否存在且存在的词频。

特征提取阶段所做的工作是词转向量和 one-hot 编码。词转向量是通过分词所应表中的分词索引列还原经过数据预处理后的句子，并用两个向量来保存数据预处理的结果。其中 X 向量是一个 63402×1 的二维向量，每一行是一条分完词的评论，Y 向量是一个 63402×1 的二维向量，每一行表示对应评论的情感倾向标签，如图 3-2 所示。

X	Y
在 2024.04.26 01:23:24 于 109ms 内执行	在 2024.04.26 01:23:27 于 14ms 内执行
'渔人',	1,
'认得',	1,
'出弹',	1,
'吉他',	1,
'嘻嘻',	1,
'天涯',	1,
'赵瑜',	1,
'小镇',	1,
'生活',	1,
'海',	1,
'故事',	1,
'秋岛',	1,
'渔人',	1,
'早安',	...]

图 3-2 X, Y 向量示例

one-hot 编码是用一个由数字组成的 $63402 \times n$ 的二维向量记录所有数据的特征，以便于模型训练。其中 n 是不重复的词语个数，由于经过停用词处理后的分词索引表有 412478 条数据，故 n 必然小于 412478。每一行表示一条评论，每一列表示某一词语，行和列交叉的每一个数据表示该分词在该评论中出现的次数。由于该二维向量中大量的数据是 0，所以用稀疏矩阵保存，只记录非零的数据。稀疏矩阵如图 3.7 所示，其中每个数据由坐标和数值，例如“(4, 36537) 2”表示第 4 行第 36537 列的值是 2，指的是第 36537 列代表的分词在第 4 条评论中出现了 2 次。经过词袋模型特征抽取后的 one-hot 编码如下图 3-3 所示：

20	# 或者将One-hot编码结果转换为稀疏矩阵的密集表示，以便更直观地查看
21	x_train_dense = x_train_one_hot.toarray()
22	print(x_train_dense[:5])
	在 2024.05.31 09:52:28 于 1s 857ms 内执行
√	(3, 27955) 1
	(3, 78681) 1
	(3, 53178) 1
	(3, 124629) 1
	(3, 40277) 1
	(3, 8360) 1
	(4, 32039) 1
	(4, 5384) 1
	(4, 36537) 2
	(4, 5127) 1
	(4, 36521) 1
	(4, 58815) 1
	(4, 19441) 1
	(4, 135830) 1

图 3-3 经过词袋模型特征抽取后的 one-hot 编码稀疏矩阵示例

3.2 词频-逆文档频率 (TF-IDF)

词频 (TF)：TF 表示词条 (关键字) 在文本中出现的频率。这个数字通常会被归一化 (一般是词频除以文章总词数)，以防止它偏向长的文件。

归一化后的词频 (TF) 计算公式记为公式 (3-1)：

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{j,k}} \quad (3-1)$$

逆文档频率 (IDF)：IDF 表示词条的普遍重要性。如果包含词条 (t_i) 的文档越少，IDF 越大，则说明词条具有很好的类别区分能力。

IDF 的计算公式记为公式 (3-2)：

$$IDF_i = \log \frac{|D|}{1 + |j: t_i \in d_j|} \quad (3-2)$$

词频-逆文档频率 (TF-IDF)：将一个词的 TF 值和 IDF 值相乘，得到该词在文档中的 TF-IDF 值，这个值代表了词在文档中的重要性。

TF-IDF 的计算公式记为公式 (3-3)：

$$TF - IDF = TF \times IDF \quad (3-3)$$

核心代码：

```
vect = TfidfVectorizer(max_df=0.5, ngram_range=(1, 1))
x_train = vect.fit_transform(x_train)
x_test = vect.transform(x_test)
```

首先，初始化 TfidfVectorizer 对象，设定参数以控制词汇表构建和特征提取过程。TfidfVectorizer 用于将文本数据转换为 TF-IDF 向量表示，即将每个文档表示为一组词项 (token) 的 TF-IDF 值构成的向量。这里指定了两个参数：

max_df=0.5：设置最高文档频率阈值。该参数限制了在所有文档中出现频率最高的那些词项。如果一个词项在超过 50% (即 0.5) 的文档中出现，那么它可能是一些语气词或者情感助词。它将被忽略 (不计入特征向量)。如果需要提取更高阶的 n-grams，可以调整此参数，如 (1, 2) 表示同时考虑单词和双词。

其次，使用 fit_transform() 方法对训练集进行 TF-IDF 转换，生成稀疏矩阵表示，并更新 TfidfVectorizer 对象的状态 (词汇表和参数)。之后，转换后的训练集表示为一个稀疏矩阵 (通常是 scipy.sparse.csr_matrix 类型)，并赋值给 x_train。

最后，使用 transform() 方法对测试集进行 TF-IDF 转换，基于训练集得到的词汇表和参数生成稀疏矩阵。并使用已经得到的词汇表和参数中的 vect 对象对测试集 x_test 进行 TF-IDF 转换。由于测试集与训练集应遵循相同的预处理规则，这里只需调用 transform() 方法。同样，x_test 会被转换为与 x_train 具有相同词汇表和特征表示形式的稀疏矩阵。

经过 TfidfVectorizer 处理后，训练集 x_train 和测试集 x_test (这里仅展示了部分训

训练集数据) 转换成的稀疏矩阵表示。训练集样本在 TF-IDF 向量化后的稀疏矩阵表示。每一行对应一个样本, 每一列对应一个词项, 元素值为该词项在对应样本中的 TF-IDF 值。例如, (0, 37221) 0.3606342836414868 表示训练集中第 0 个样本中, 词汇表中索引为 37221 的词项的 TF-IDF 值为 0.3606。部分经过 TF-IDF 特征抽取之后的数据如下图 3-4 所示:

构建决策树模型 (使用 TF-IDF 特征) 。。	
(0, 37221)	0.3606342836414868
(0, 119788)	0.6058504376356686
(0, 65861)	0.49278360600268717
(0, 378)	0.5099534080031711
(1, 103558)	0.34955323164926627
(1, 94073)	0.4257085508880638
(1, 122620)	0.2621479707721354
(1, 63578)	0.552597435050421

图 3-4 各特征对应 TF-IDF 值

3.3 Word2Vec

Word2Vec 是从大量文本语料中以有监督的方式学习语义知识的一种模型, 它被大量地用在自然语言处理 (NLP) 中。Word2Vec 其实就是通过学习文本特征并用词向量的方式表征词的语义信息, 即通过一个嵌入空间使得语义上相似的单词在该空间内距离很近。Embedding 其实就是一个映射, 将单词从原先所属的空间映射到新的多维空间中, 也就是把原始词汇所在的空间嵌入到一个新的词汇空间中去。

在自然语言处理领域, 词向量是一种强大的工具, 它将词汇编码为数值形式, 从而使得计算机能够理解和处理语言。例如, 考虑词汇 “cat” 和 “kitty”, 这两个词在语义上具有高度的相似性。相比之下, “dog” 虽然也属于宠物类别, 但与 “kitty” 在语义上的联系就没有那么紧密。至于 “apple”, 它与 “kitty” 之间的关系就更加遥远, 仿佛完全没有联系。通过将词汇转换为词向量, 词嵌入模型赋予了每个词在多维空间中的坐标, 这让机器能够利用数学运算来探索词汇之间的关系。

在 Skip-gram 模型中, 目标是给定中心词 ω_c , 预测其上下文词 ω_o 。为了实现这一点, 模型定义了一个条件概率分布 $P(\omega_o | \omega_c)$, 表示在给定中心词 ω_c 的情况下, 观察到上下文词 ω_o 的概率。在实际训练中, 算法会最大化所有有效词对 (中心词-上下文词对) 的联合概率。

对于词向量表示, 每个词 ω 都被映射为一个固定长度的向量, 记作 v_ω (对于中心词) 和 v_ω (对于上下文词)。Skip-gram 模型通过 softmax 函数和一个得分函数来估计条件概率记为公式 (3-4):

$$P(\omega_o \mid \omega_c) = \frac{\exp(v_{\omega_o} \cdot v_{\omega_c})}{\sum_{\omega'_o \in V} \exp(v_{\omega'_o} \cdot v_{\omega_c})} \quad (3-4)$$

其中, v_{ω_o} 是上下文词 ω_o 的向量表示, v_{ω_c} 是中心词 ω_c 的向量表示, V 是词汇表, $v_{\omega_o} \cdot v_{\omega_c}$ 表示两个向量的点积, 它衡量了两个词在嵌入空间中的相似度。

总结起来,在 Word2vec 模型中,通过优化上述条件概率或其近似变体的损失函数,模型能够在训练过程中不断调整词向量,使得语义上相关的词在高维空间中的向量更加接近,从而实现高效且有意义的词嵌入。通过 `build_word2vec_model` 函数所指定的参数,参数可以控制词向量的维度、上下文窗口大小以及最小词频阈值等,以便适应不同的应用场景和数据集特点,进而得到适用于下游 NLP 任务的高质量词向量表示。

通过初始化、构建词汇表、训练这三个步骤构建并训练一个 Word2Vec 模型，最后返回训练好的模型供后续使用。这个模型能够将输入文本中的词语映射到向量空间中，形成具有语义信息的词向量。使用数据集 'weibo_senti_100k.csv' 训练生成的 100 维 word2vec 模型 “word2vec.txt” 如下图 3-5 所示：



图 3-5 word2vec.txt

第4章 新闻评论情感分析算法实现

4.1 数据集划分

经过获取新闻评论数据、预处理、特征抽取三个过程之后，即可使用不同的分类算法训练新闻评论情感分析模型。在模型训练之前，需要先对数据集进行划分。数据集划分的目的是将特征提取后的数据分为训练集和测试集，训练集的数据用于训练模型，模型训练完成后用测试集进行测试。训练集的数据规模往往大于测试集的数据规模，在本文中，80%的数据分到训练集用于训练模型，20%的数据分到测试集用于测试模型。

划分数据集之后，分别使用决策树（4.2）、朴素贝叶斯（4.3）、支持向量机（4.4）三种传统机器学习算法进行文本情感分类。通过不同的特征工程和分类算法进行新闻评论分类，计算不同特征工程和分类算法的准确度，讨论不同方法的优劣。

4.2 基于决策树分类算法的新闻评论情感分析

4.2.1 训练模型

决策树模型，其基本原理来自香浓所提的信息论。在信息论中，信息被定义为消除随机不确定性的东西，通常用信息熵来衡量，单位为比特。

信息熵的计算公式如式（4-1）所示。其中 $H(X)$ 表示信息熵， x 表示随机变量， $P(x)$ 表示概率输出函数，式中对数一般取 2 为底。

$$H(X) = -\sum_{i=1}^n P(x_i) \log_b P(x_i) \quad (4-1)$$

决策树依据信息增益来进行构建，信息增益的计算公式如式（4-2）所示。其中 X 表示随机变量， A 表示特征， $g(X, A)$ 表示信息增益记为公式（4-2）， $H(X)$ 表示信息熵， $H(X|A)$ 表示条件熵。划分决策树时，某一特征的信息增益越大，该特征所占的比重就越大。

$$g(X, A) = H(X) - H(X|A) \quad (4-2)$$

决策树模型是一种常用的机器学习方法，它模拟人类决策过程来预测数据的分类或数值。决策树由节点和分支构成，其中每个内部节点代表一个属性上的测试，每个分支代表测试的结果，而每个叶节点代表一个类别或数值。

构建决策树的有以下几步：

步骤一，选择最佳分裂属性：使用算法（如信息增益、增益比或基尼不纯度）来选择一个属性作为节点分裂的依据。

步骤二，分裂节点：根据选定的属性分裂节点，生成子节点。

步骤三，递归构建：对每个子节点重复步骤 1 和 2，直到满足停止条件（如节点包含的样本数小于阈值、节点纯度达到一定程度或达到预设的树深度）。

决策树分类算法伪代码如算法 1 所示：

算法 1：决策树

输入：训练集 D ，待分类文本 X

输出：文本 X 的预测情感类别 P

1. *Init*: 构建文本表示模型，计算训练集 D 中每个词的信息增益
2. *构建决策树*:
3. for 训练集 D 中的每个特征 f do
4. 使用信息增益选择最佳分裂特征
5. 根据最佳特征分裂训练集，创建子节点
6. end for
7. if 子节点纯净或达到预设条件 then
8. 标记为叶节点，赋予类别标签
9. else
10. 递归构建子树
11. end if
12. end *构建决策树*
13. for 文本 X 中的每个词 W do
14. if W 在文本表示模型中 then
15. 沿着决策树传递 w ，直到达到叶节点
16. end if
17. end for
18. P = 达到的叶节点的类别标签
19. return P

4.2.2 模型测试

模型测试是指在建立统计模型、机器学习模型或任何预测性数据分析模型后，对其性能和可靠性进行评估的过程。这个过程旨在确保模型能够准确地泛化到未见过的数据，而不仅仅是拟合训练数据。模型验证的主要目标是检测过拟合（即模型在训练数据上表现优秀，但在新数据上表现较差的情况）、评估模型的稳定性和可靠性，并最终确定模

型是否适用于其预期目的。

本文主要模型调优步骤有以下两点：

(1) 交叉验证：这是一种常用的技术，用于更可靠地估计模型的性能。常见的交叉验证方法包括 k 折交叉验证，其中数据被分成 k 个子集，每次迭代时用 k-1 个子集训练模型，剩下的一个子集用来验证，这个过程重复 k 次，每次变换验证集，最后求平均性能指标。

(2) 超参数调优：基于验证集的结果调整模型的超参数，以优化模型性能。

模型测试时，使用模型预测测试集的数据，计算预测结果的准确度作为测试结果，用于评价模型。测试模型时，最重要的评价指标为准确率 (*Accuracy*) 记为公式 (4-3)：

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (4-3)$$

为进一步度量评估结果的有效性，实验采用精确率 (*Precision*)、召回率 (*Recall*) 和 F1 值 (*F1-score*) 三个指标对其进行评价，其计算方法分别如公式 (4-4)、公式 (4-5) 和公式 (4-6) 所示，计算用到的列联表如表 4-1 所示。

$$Precision = \frac{TP}{TP+FP} \quad (4-4)$$

$$Recall = \frac{TP}{TP+FN} \quad (4-5)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4-6)$$

人工验证主要使用人工对标注数据的验证，以评估模型的准确性。在本论文中，主要使用人工对新闻新闻评论情感分析机器学习模型输出的预测结果的准确率进行计算。计算公式如公式 (4-7) 所示。

$$人工验证准确率 = \frac{TP}{TP+FP} \quad (4-7)$$

在机器学习分类模型中，精确率是指模型预测为正类的样本中，实际上是正类的比例。它反映了模型的查准率，即正确预测为正类的能力。精确率越高，将负类误判为正类的可能性就越小。召回率是指实际为正类的样本中，模型成功预测为正类的比例。它反映了模型的查全率，即找全正类的能力。召回率越高，将正类漏判的可能性就越小。F1 值综合了精确率和召回率，是两者的加权调和平均。它是一个综合性能指标，用于衡量模型的整体效果。F1 值越高，说明模型在平衡查准率和查全率方面表现更好。

表 4-1 计算机器学习分类模型描述结果 P, R 和 F 值的列联表

	算法标注为正类	算法标注为负类
人工标注为正类	TP	FN
人工标注为负类	FP	TN

4.2.3 词袋模型的决策树模型调优及保存

对决策树模型准确率影响最大的一个参数是决策树的深度，即构建决策树时所允许的最大深度，这是该模型的一个需要手动调整的超参数。深度越大，模型越复杂，模型能够学习到更细致的数据特征和更复杂的决策边界，容易出现过拟合的情况，模型在训练集表现良好，从而在新的、未见过的数据上表现不佳。而深度越小，模型越简单，容易出现欠拟合的情况，模型不能很好地学习特征，对于人工测试时准确率表现不佳。因此，决策树的深度需要通过交叉验证等方法来仔细调整，以找到既不过拟合也不欠拟合的最佳平衡点。这样的模型既能够捕捉到数据中的基本规律，又能够抵抗训练数据中的随机噪声，从而在实际应用中达到较高的准确率。

每一次调整完决策树的深度，都需要重新进行模型测试，计算模型的各种评价指标。需要反复优化与测试，找到一个最佳的决策树深度，作为最终的新闻评论情感分析模型。23-29 层决策树各准确率如下表 4-2 所示：

表 4-2 词袋模型不同决策树深度的模型准确率

决策树深度	准确率
33	0.80522
34	0.80559
35	0.80659
36	0.80710
37	0.80743
38	0.80676
39	0.80643

通过观察发现 33-39 层准确率基本保持在 80%左右，系统自动选择 37 层作为最佳决策树深度，构建最终词袋模型的决策树模型，最终词袋模型配合决策树分类算法在随机测试集中的消极情绪预测准确率为 75%，积极情绪预测准确率为 89%。最终模型的性能指标如下表 4-3 所示：

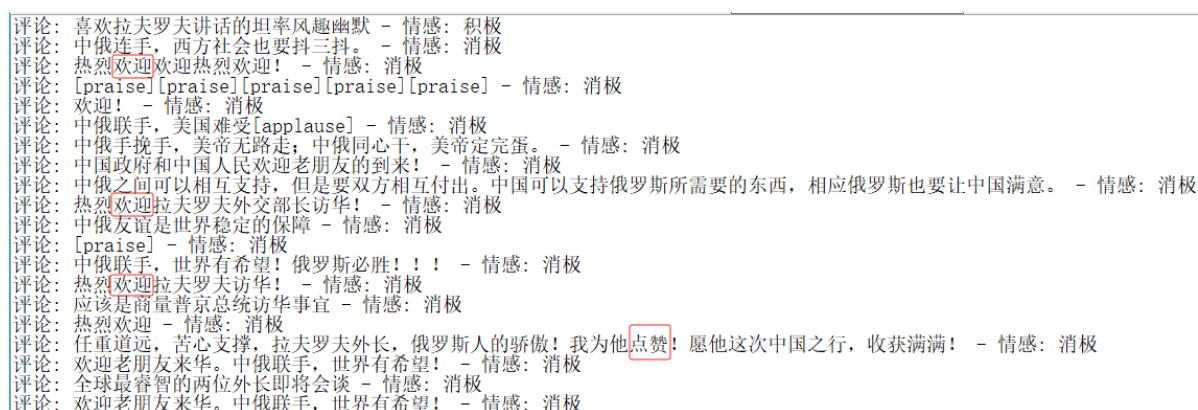
表 4-3 最终词袋模型的决策树模型的性能指标

	precision	recall	f1-score	support
0	0.76	0.90	0.82	11940
1	0.88	0.71	0.79	12020
accuracy			0.81	23960
macro avg	0.82	0.81	0.81	23960
weighted avg	0.82	0.81	0.81	23960

在机器学习项目中，模型和向量器的保存是一个重要的步骤。这个过程将经过训练和调整的模型以及用于转换数据的向量器导出并存储到本地文件系统中。这样做的目的

是为了能够在未来的数据分析任务中可以直接使用这些模型和向量器，而无需重新耗时去训练新闻评论情感分析模型。该“词袋模型的决策树”新闻评论情感分析模型在预测消极情感时的精确率为 75%，而在预测积极情感时的精确率高达 89%。这种显著的精确率差异揭示了模型在情感分析上的一个潜在问题，即存在“消极偏见”。也就是说，模型在识别消极情感方面的表现不如在识别积极情感方面，这可能导致对消极评论的过度敏感或对积极评论的不足关注。

在系统中，使用已经保存好的词袋模型和决策树模型进行新闻评论的情感分析。如图 4-1 所示：



```

评论：喜欢拉夫罗夫讲话的坦率风趣幽默 - 情感：积极
评论：中俄联手，西方社会也要抖三抖。 - 情感：消极
评论：热烈欢迎欢迎热烈欢迎！ - 情感：消极
评论：[praise][praise][praise][praise][praise] - 情感：消极
评论：欢迎！ - 情感：消极
评论：中俄联手，美国难受[applause] - 情感：消极
评论：中俄手挽手，美帝无路走；中俄同心干，美帝定完蛋。 - 情感：消极
评论：中国政府 and 中国人民欢迎老朋友的到来！ - 情感：消极
评论：中俄之间可以相互支持，但是要双方相互付出。中国可以支持俄罗斯所需要的东西，相应俄罗斯也要让中国满意。 - 情感：消极
评论：热烈欢迎拉夫罗夫外交部长访华！ - 情感：消极
评论：中俄友谊是世界稳定的保障 - 情感：消极
评论：[praise] - 情感：消极
评论：中俄联手，世界有希望！俄罗斯必胜！！ - 情感：消极
评论：热烈欢迎拉夫罗夫访华！ - 情感：消极
评论：应该是商量普京总统访华事宜 - 情感：消极
评论：热烈欢迎 - 情感：消极
评论：任重道远，苦心支撑，拉夫罗夫外长，俄罗斯人的骄傲！我为他点赞！愿他这次中国之行，收获满满！ - 情感：消极
评论：欢迎老朋友来华。中俄联手，世界有希望！ - 情感：消极
评论：全球最睿智的两位外长即将会谈 - 情感：消极
评论：欢迎老朋友来华。中俄联手，世界有希望！ - 情感：消极
    
```

图 4-1 词袋模型决策树模型进行新闻评论情感分析

在进行了对凤凰新闻用户评论数据的详尽分析之后，我们发现了一个引人注目的问题：模型在情感分析的过程中未能准确地识别和解读某些关键词的情感色彩。具体来说，一些通常具有积极含义的词汇，如“欢迎”和“点赞”，以及一些倾向于表达消极情绪的词语，如“完蛋”和“也就那样”，都没有被模型正确地分类。这种现象在人工测试的 157 条评论数据中表现得尤为明显，其中只有 69 条评论被模型正确地预测了情感倾向，这意味着模型的准确率大约只有 43.9%。

这样的测试结果说明了模型在情感分析方面的一个显著偏差，特别是它对消极情绪的偏见。此外，这一结果还暗示了模型可能存在过拟合的问题。过拟合是指模型在训练集上的表现异常出色，以至于它学习到了训练数据中的特定噪声和细节，而不是能够代表整体数据分布的潜在规律。因此，当模型面对新的、未见过的数据时，它的预测性能会大打折扣。在这种情况下，模型虽然能够在训练集上达到高准确率，但在实际应用中，尤其是在处理与训练集分布不同的数据时，它的泛化能力却显得不足。

4.2.4 TF-IDF 算法的决策树模型调优及保存

方法同 4.2.3，决策树的深度使用交叉验证等方法来进行调整，以找到既不过拟合也不欠拟合的最佳平衡点。这样的模型既能够捕捉到数据中的基本规律，又能够抵抗训练数据中的随机噪声，从而在实际应用中达到较高的准确率。每一次调整完决策树的深度，都需要重新进行模型测试，计算模型的准确度。反复优化与测试，找到一个最佳的决策

树深度，作为最终的模型。52-58 层决策树各准确率如下表 4-4 所示：

表 4-4 TF-IDF 不同决策树深度的模型准确率

决策树深度	准确率
52	0.79765
53	0.79873
54	0.79948
55	0.80044
56	0.79961
57	0.80028
58	0.80032

通过观察发现 52-58 层准确率基本保持在 80%左右，系统自动选取 55 层作为最佳决策树深度，构建最终 TF-IDF 的决策树模型，最终 TF-IDF 算法配合决策树分类算法在随机测试集中的消极情绪预测精确率为 75%，积极情绪预测精确率为 87%。最终准确率模型的性能指标如下表 4-5 所示：

表 4-5 第 55 层 TF-IDF 的决策树模型的性能指标

	precision	recall	f1-score	support
0	0.75	0.90	0.82	11949
1	0.87	0.70	0.78	12049
accuracy			0.80	23998
macro avg	0.81	0.80	0.80	23998
weighted avg	0.81	0.80	0.80	23998

模型和向量器保存是将已经训练好的模型和向量器保存到本地文件，模型最终通过自动划分的测试机来测试发现消极情感预精确率率在 75%，积极情感预测精确率在 81%，消极和积极情感预测精确率差别过大，也无法很好地对陌生新闻评论数据集进行情感分析。这种显著的精确率差异揭示了模型在情感分析上的一个潜在问题，即存在“消极偏见”。也就是说，模型在识别消极情感方面的表现不如在识别积极情感方面，这可能导致对消极评论的过度敏感或对积极评论的不足关注。

在系统中使用已经保存好的 TF-IDF 决策树模型进行新闻评论情感分析如图 4-2 所示：

评论: 喜欢拉夫罗夫讲话的坦率风趣幽默 - 情感: 消极
 评论: 中俄联手, 西方社会也要抖三抖。 - 情感: 消极
 评论: 热烈欢迎欢迎热烈欢迎! - 情感: 消极
 评论: [praise][praise][praise][praise][praise] - 情感: 消极
 评论: 欢迎! - 情感: 消极
 评论: 中俄联手, 美国难受[applause] - 情感: 消极
 评论: 中俄手挽手, 美帝无路走; 中俄同心干, 美帝定完蛋。 - 情感: 消极
 评论: 中国政府和中国人民欢迎老朋友的到来! - 情感: 消极
 评论: 中俄之间可以相互支持, 但是要双方相互付出。中国可以支持俄罗斯所需要的东西, 相应俄罗斯也要让中国满意。 - 情感: 消极
 评论: 热烈欢迎拉夫罗夫外交部长访华! - 情感: 消极
 评论: 中俄友谊是世界稳定的保障 - 情感: 消极
 评论: [praise] - 情感: 消极
 评论: 中俄联手, 世界有希望! 俄罗斯必胜!!! - 情感: 消极
 评论: 热烈欢迎拉夫罗夫访华! - 情感: 消极
 评论: 应该是商量普京总统访华事宜 - 情感: 消极
 评论: 热烈欢迎 - 情感: 消极
 评论: 任重道远, 苦心支撑, 拉夫罗夫外长, 俄罗斯人的骄傲! 我为他点赞! 愿他这次中国之行, 收获满满! - 情感: 消极
 评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极
 评论: 全球最睿智的两位外长即将会谈 - 情感: 消极
 评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极

图 4-2 TF-IDF 决策树模型进行新闻评论情感分析

在深入分析凤凰新闻用户评论数据后, 发现了模型在情感分析任务中的一些显著不足。模型未能准确识别包含“欢迎”、“点赞”、“希望”和“风趣幽默”等词汇的评论为积极情绪, 同时也未能正确判断“完蛋”等词汇所表达的消极情绪。这种情感识别的不准确性在人工测试 157 条数据时得到了进一步的证实, 其中模型只有 65 条预测是正确的, 导致整体准确率仅为 41.4%。这不仅表明模型有一个明显的消极偏向, 而且还存在过拟合的问题, 即模型在训练集上的表现远远超过了其在未知数据上的泛化能力。

4.3 基于朴素贝叶斯分类算法的新闻评论情感分析

4.3.1 训练模型

朴素贝叶斯分类器基于贝叶斯定理和特征条件独立假设进行工作。它的核心公式是贝叶斯定理, 用于计算在给定观测数据条件下某个类别的后验概率。对于分类问题, 朴素贝叶斯公式表述为: 对于给定的观测数据 $X = (x_1, x_2, x_3, \dots, x_n)$ 和类别 y , 计算 y 为某一特定类 c 的后验概率记为公式 (4-8):

$$P(y = c|x) = \frac{P(X|y=c)P(y=c)}{P(x)} \quad (4-8)$$

其中: $P(y=c|x)$ 是在给定观测数据 X 的条件下, 类别 y 为 c 的后验概率, 这是项目想要的目标。

$P(x|y=c)$ 是在类别为 c 的条件下, 观测数据 x 出现的概率, 也称为似然概率 (likelihood)。

$P(y=c)$ 是类别 c 的先验概率, 即在未观察到数据 x 之前, 类别 c 自身的概率。

$P(x)$ 是观测数据 x 的边缘概率 (marginal probability), 通常作为归一化因子, 确保后验概率之和为 1。在实际应用中, 由于 $P(x)$ 对于所有类别 c 是常数, 因此在比较不同类别 c 的后验概率以进行分类时, 往往可以忽略。朴素贝叶斯分类器记为公式 (4-9) 的“朴素”体现在它假设特征之间是条件独立的, 即:

$$P(X|y = c) = \prod_{i=1}^n P(x_i|y = c) \quad (4-9)$$

这意味着在给定类别 c 的条件下, 每个特征 x_i 的出现独立于其他特征。这一假设简

化了计算，使得模型在有限数据下仍能有效工作，尽管在现实世界中特征之间可能存在某种程度的相关性。在实际应用朴素贝叶斯分类器时，通常会先估计每个类别的先验概率 $P(y = c)$ 以及每个特征 x_i 在各类别下的条件概率 $P(x_i | y = c)$ ，然后使用这些概率值和给定的观测数据 x 来计算类别 c 的后验概率 $P(y = c | x)$ 。分类时，选择后验概率记为公式（4-10）最高的类别作为预测结果：

$$\hat{y} = \underset{c}{\operatorname{argmax}} P(y = c | x) \quad (4-10)$$

朴素贝叶斯分类器的公式基于贝叶斯定理，并利用特征条件独立假设简化了计算，通过估计和利用先验概率及条件概率来确定给定观测数据最可能的类别。

朴素贝叶斯分类算法伪代码如算法 2 所示：

算法 2：朴素贝叶斯

输入：训练集 D ，待分类的文本 X

输出：文本 X 的预测情感类别 P

1. *Init*: 从训练集 D 中计算每个词的情感倾向分数
 2. 构建朴素贝叶斯:
 3. for 文本 X 中的每个词 w do
 4. if w 在文本表示模型中 then
 5. 计算 w 的情感倾向分数
 6. end if
 7. end for
 8. for 训练集 D 中的每个类别 c do
 9. 计算类别 c 的先验概率和条件概率
 10. end for
 11. for 文本 X 中的每个词 w do
 12. 使用朴素贝叶斯公式计算文本 X 属于每个类别 c 的概率
 13. end for
 14. end 构建朴素贝叶斯
 15. P = 概率最高的类别
 16. return P
-

4.3.2 模型测试

如第 4.2.2 节所述，本小结研究采用了相同的模型测试方法。

4.3.3 TF-IDF 算法的朴素贝叶斯模型调优及保存

在朴素贝叶斯（MultinomialNB）模型调优中，网格搜索（Grid Search）是一种常用的方法，它通过遍历所有可能的参数组合来寻找最佳的模型配置。这个过程通常涉及到多个随机种子（random_state）的设定，以确保训练集和测试集的划分能够全面反映模型在不同数据分布下的性能。每个随机种子下，模型都会进行交叉验证，这样可以在不同的数据子集上评估模型的稳定性和准确性。最终，通过比较不同随机种子下的交叉验证得分，可以选择出在多数情况下表现最佳的模型参数。这种方法不仅提高了模型的泛化能力，也使得模型在实际应用中更加可靠。

最后，找出所有随机种子下的最佳得分和参数，使用这些参数训练模型并在测试集上评估模型准确率，以评估模型的稳定性和泛化能力，每一次调整模型参数，都需要重新进行模型测试，计算模型的准确度。反复优化与测试，找到一个最佳参数，构建最终的朴素贝叶斯模型。不同参数朴素贝叶斯模型各准确率如下表 4-6 所示：

表 4-6 不同随机种子 TF-IDF 的朴素贝叶斯模型各准确率

不同随机种子	最佳交叉验证准确率
42	0.7995
52	0.8006
62	0.8010
72	0.7984
82	0.7989

通过观察发现不同参数准确率基本保持在 80%左右，系统自动选取最佳随机种子值为 62，构建最终 TF-IDF 的朴素贝叶斯模型。最终准确率模型的性能指标如下表 4-7 所示：

表 4-7 TF-IDF 的朴素贝叶斯模型的性能指标

	precision	recall	f1-score	support
0	0.81	0.79	0.80	12059
1	0.79	0.82	0.80	11939
accuracy			0.80	23998
macro avg	0.80	0.80	0.80	23998
weighted avg	0.80	0.80	0.80	23998

模型和向量器保存是将已经训练好的模型和向量器保存到本地文件，模型最终通过自动划分的测试机来测试发现消极情感预测精确率在 81%，积极情感预测精确率在 79%，

消极和积极情感预测精确率差别已经变小，已经基本具有对陌生新闻评论数据集进行预测的能力了。在系统中使用已经保存好的 TF-IDF 朴素贝叶斯模型进行新闻评论情感分析如图 4-3 所示：

```

评论: 喜欢拉夫罗夫讲话的坦率风趣幽默 - 情感: 积极
评论: 中俄联手, 西方社会也要抖三抖。 - 情感: 消极
评论: 热烈欢迎欢迎热烈欢迎! - 情感: 积极
评论: [praise][praise][praise][praise][praise] - 情感: 消极
评论: 欢迎! - 情感: 消极
评论: 中俄联手, 美国难受[applause] - 情感: 消极
评论: 中俄手挽手, 美帝无路走; 中俄同心干, 美帝定完蛋。 - 情感: 消极
评论: 中国政府和中国人民欢迎老朋友的到来! - 情感: 积极
评论: 中俄之间可以相互支持, 但是要双方相互付出。中国可以支持俄罗斯所需要的东西, 相应俄罗斯也要让中国满意。 - 情感: 积极
评论: 热烈欢迎拉夫罗夫外交部长访华! - 情感: 积极
评论: 中俄友谊是世界稳定的保障 - 情感: 积极
评论: [praise] - 情感: 消极
评论: 中俄联手, 世界有希望! 俄罗斯必胜!!! - 情感: 积极
评论: 热烈欢迎拉夫罗夫访华! - 情感: 积极
评论: 应该是商量普京总统访华事宜 - 情感: 积极
评论: 热烈欢迎 - 情感: 积极
评论: 任重道远, 苦心支撑, 拉夫罗夫外长, 俄罗斯人的骄傲! 我为他点赞! 愿他这次中国之行, 收获满满! - 情感: 积极
评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 积极
评论: 全球最睿智的两位外长即将会谈 - 情感: 积极
评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 积极
    
```

图 4-3 TF-IDF 朴素贝叶斯模型进行新闻评论情感分析

通过对比验证发现，该模型正确率已经提高并且已经消极和积极识别能力已经增强。最终人工测试 157 条数据，模型输出的预测只有 92 条分析正确。推算最终的模型的准确率大概在 58.5% 左右。单一情感倾向的问题已经缓解。但是诸如“欢迎”“希望”一些关键词还是无法正常识别。该模型还是无法对一些长文本，需要联系上下文的新闻评论做出很好的情感识别，所以考虑使用其他传统机器学习算法和特征工程的组合。

4.4 基于支持向量机分类算法的新闻评论情感分析

4.4.1 训练模型

支持向量机 (Support Vector Machine, SVM) 是一种强大的监督学习算法，主要用于分类任务，也可以通过核技巧扩展到非线性分类以及回归分析。SVM 的核心思想是在高维特征空间中构建一个最优的决策边界（超平面），使得两类样本尽可能被准确且最大间距地分开。在二维或者三维空间中，决策边界是一个线或者面，而在更高维度的空间中则是一个超平面。

由于中文文本情感分类大部分情况下是非线性的分类问题。虽然理论上可以尝试构建线性模型来进行分类，但自然语言的特点决定了文本数据在高层次语义上常常呈现出复杂的非线性关系。例如，情感倾向不仅取决于单个词汇或短语，还受到上下文、句法结构、修辞手法等多种因素的影响，这些影响很难用一个简单的线性模型捕捉，这就需要引入非线性技术。

非线性支持向量机 (Nonlinear SVM) 是一种扩展自标准线性支持向量机的机器学习模型，旨在处理非线性可分的数据集。对于非线性可分的数据，本文通过使用核函数 $k(x_i, x_j)$ 将低维数据映射到高维特征空间，常用的核函数有径向基函数 (RBF) 核、多项式核等。优化问题保持不变，但在内积 $w \cdot x$ 处替换为核函数计算记为公式 (4-11)，

$\phi(x_i)$ 是将输入数据映射到高维特征空间的映射函数，映射函数记为公式（4-12）：

$$y_i(\omega \cdot \phi(x_i) + b) \geq 1 - \xi_i \quad (4-11)$$

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (4-12)$$

支持向量机分类算法伪代码如算法 3 所示：

算法 3：支持向量机

输入：训练集 D ，待分类的文本 X

输出：文本 X 的预测情感类别 P

1. *Init*: 使用文本表示模型初始化 SVM 分类器
 2. 构建支持向量机:
 3. for 训练集 D 中的每个样本 d do
 4. 将文档 d 转换为特征向量 v
 5. end for
 6. 使用特征向量 v 训练 SVM 分类器
 7. 将待分类文本 X 转换为特征向量 v_x
 8. 使用 SVM 分类器对特征向量 v_x 进行分类
 9. end 构建支持向量机
 10. P = 分类结果
 11. return P
-

4.4.2 模型测试

如第 4.2.2 节所述，本小结研究采用了相同的模型测试方法。

4.4.3 TF-IDF 算法的支持向量机模型调优及保存

Scikit-learn 库提供了一个高效的工具来实现 SVM 模型，其中包括对模型的超参数进行优化的功能。超参数优化是一个重要的步骤，因为它可以显著提高模型的性能。在 SVM 模型中，特别是当使用非线性核函数，如径向基函数（RBF 核），两个关键的超参数是 C 和 γ 。 C 参数是正则化参数，它可以帮助模型在训练数据的拟合度和泛化能力之间找到平衡。 C 值较小会导致模型有更高的偏差和较低的方差，而 C 值较大则会导致模型有更低的偏差和较高的方差。另一方面， γ 参数控制着 RBF 核的宽度，从而影响到决策边界的形状。 γ 值较小意味着决策边界会更加平滑，模型的影响范围更广，而较大的 γ 值会使得决策边界更加陡峭，模型更加关注于接近支持向量

的数据点。

为了找到这两个参数的最佳组合，我们通常会定义一个参数网格 `param_grid`，它包含了 `C` 和 `gamma` 的不同取值组合。然后，我们可以使用 `Scikit-learn` 中的网格搜索（`Grid Search`）来遍历这些组合，通过交叉验证来评估每一组参数的性能。最终，我们会选择那些在交叉验证中表现最好的参数组合，以期望模型在未见过的数据上也能有良好的表现。不同超参数 `C` 和 `gamma` 的取值和对应的准确率如下表 4-8 所示：

表 4-8 不同超参数 `C` 和 `gamma` 的 TF-IDF 算法的支持向量机准确率

超参数 <code>C</code>	<code>gamma</code>	平均交叉验证准确率
0.1	0.01	0.6863
0.1	0.1	0.7830
0.1	1	0.7983
1	0.01	0.7871
1	0.1	0.8228
1	1	0.8302
10	0.01	0.8244
10	0.1	0.8196
10	1	0.8254

系统自动选取模型交叉验证准确率最高的参数为 `C=1`，`Gamma=1`，平均交叉验证准确率为 83%，根据这两个参数构建最终 TF-IDF-SVM 模型，最终模型的性能指标如下表 4-9 所示：

表 4-9 最终 TF-IDF 算法的支持向量机的性能指标

	precision	recall	f1-score	support
0	0.81	0.87	0.84	11949
1	0.86	0.80	0.82	12049
accuracy			0.83	23998
macro avg	0.83	0.83	0.83	23998
weighted avg	0.83	0.83	0.83	23998

通过观察可知，该模型在负面精确率为 81%，正面精确率为 86%，相较于之前的分类算法。精确率已经显著提高，虽然正负面精确率差距也在变小，但是并没有缩小到理想范围内，本文使用陌生新闻评论数据集也就是直接使用系统爬取在线凤凰新闻用户评论数据，进行人工估计准确率，系统中使用 TF-IDF 算法的支持向量机模型进行用户评论情感分析的页面如下图 4-4 所示：

评论: 喜欢拉夫罗夫讲话的坦率风趣幽默 - 情感: 积极
 评论: 中俄联手, 西方社会也要抖三抖。 - 情感: 消极
 评论: 热烈欢迎热烈欢迎! - 情感: 积极
 评论: [praise][praise][praise][praise][praise] - 情感: 消极
 评论: 欢迎! - 情感: 消极
 评论: 中俄联手, 美国难受[applause] - 情感: 消极
 评论: 中俄手挽手, 美帝无路走; 中俄同心干, 美帝定完蛋。 - 情感: 消极
 评论: 中国政府和中国人民欢迎老朋友的到来! - 情感: 积极
 评论: 中俄之间可以相互支持, 但是要双方相互付出。中国可以支持俄罗斯所需要的东西, 相应俄罗斯也要让中国满意。 - 情感: 积极
 评论: 热烈欢迎拉夫罗夫外交部长访华! - 情感: 积极
 评论: 中俄友谊是世界稳定的保障 - 情感: 消极
 评论: [praise] - 情感: 消极
 评论: 中俄联手, 世界有希望! 俄罗斯必胜!!! - 情感: 消极
 评论: 热烈欢迎拉夫罗夫访华! - 情感: 积极
 评论: 应该是商量普京总统访华事宜 - 情感: 消极
 评论: 热烈欢迎 - 情感: 积极
 评论: 任重道远, 苦心支撑, 拉夫罗夫外长, 俄罗斯人的骄傲! 我为他点赞! 愿他这次中国之行, 收获满满! - 情感: 积极
 评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极
 评论: 全球最睿智的两位外长即将会谈 - 情感: 积极
 评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极

图 4-4 系统中使用 TF-IDF 算法的支持向量机模型

经过对比系统中输出的分析结果, 发现还是有一些词语没办法很好地识别其中的情感, 例如“欢迎”“有希望”, 也无法联系上下文进行识别。最终人工测试 157 条数据, 模型输出的预测已有 98 条分析正确。推算最终的模型的准确率大概在 62.4%左右。依旧无法对上下文和有联系的关键词进行识别。由于无法联系上下文是一个重大问题。所以考虑使用其他特征工程来配合 SVM 分类算法进行分析。

4.4.4 Word2vec 算法的支持向量机模型调优及保存

使用 scikit-learn 库进行机器学习模型参数调优, 特别是针对非线性核函数(RBF 核)的 C 和 gamma 参数。创建一个 GridSearchCV 交叉验证网格搜索工具, 遍历参数网格中的每个组合, 并使用交叉验证方案评估模型性能。这里选择 SVM 分类器, 设置内核类型为 'rbf', 即将使用 RBF 非线性核函数, 同时设置了 cv=3, 这意味着将使用 3 折交叉验证来评估每个参数组合。

最后, 指定超参数 C 和 gamma 参数。使用这些参数训练模型并在测试集上评估模型准确率, 以评估模型的稳定性和泛化能力。不同超参数 C 和 gamma 参数的 Word2vec 算法的支持向量机模型各准确率如下表 4-10 所示:

表 4-10 不同超参数 C 的 Word2vec 算法的支持向量机准确率

超参数 C	gamma	平均交叉验证准确率
0.1	0.01	0.8532
0.1	0.1	0.8709
0.1	1	0.8120
1	0.01	0.8728
1	0.1	0.8837
1	1	0.8624
10	0.01	0.8822
10	0.1	0.8864
10	1	0.8510

最终系统自动选定最优参数 C 为 10, Gamma 为 0.1, 最终 Word2vec 算法的支持向量机的性能指标如下表 4-11 所示:

表 4-11 最终 Word2vec 算法的支持向量机的性能指标

	precision	recall	f1-score	support
0	0.91	0.86	0.89	11949
1	0.87	0.92	0.89	12049
accuracy			0.89	23998
macro avg	0.89	0.89	0.89	23998
weighted avg	0.89	0.89	0.89	23998

模型最终通过自动划分的测试机来测试发现消极情感预测精确率在 92%, 积极情感预测精确率在 85%, 消极和积极情感预测精确率差别很小, 并且整体准确率也较高, 在系统中使用 Word2vec 算法支持向量机模型进行新闻评论情感分析如图 4-5 所示:

评论: 喜欢拉夫罗夫讲话的坦率[风趣幽默] - 情感: 积极
评论: 中俄联手, 西方社会也要抖三抖。 - 情感: 消极
评论: 热烈欢迎欢迎热烈欢迎! - 情感: 积极
评论: [praise][praise][praise][praise][praise] - 情感: 积极
评论: 欢迎! - 情感: 积极
评论: 中俄联手, 美国难受[applause] - 情感: 消极
评论: 中俄手挽手, 美帝无路走; 中俄同心干, 美帝定完蛋。 - 情感: 积极
评论: 中国政府和中国人民欢迎老朋友的到来! - 情感: 积极
评论: 中俄之间可以相互支持, 但是要双方相互付出。中国可以支持俄罗斯所需要的东西, 相应俄罗斯也要让中国满意。 - 情感: 积极
评论: 热烈欢迎拉夫罗夫外长部长访华! - 情感: 积极
评论: 中俄友谊是世界稳定的保障 - 情感: 消极
评论: [praise] - 情感: 积极
评论: 中俄联手, 世界有希望! 俄罗斯必胜!!! - 情感: 消极
评论: 热烈欢迎拉夫罗夫访华! - 情感: 积极
评论: 应该是商量普京总统访华事宜 - 情感: 积极
评论: 热烈欢迎 - 情感: 积极
评论: 任重道远, 苦心支撑, 拉夫罗夫外长, 俄罗斯人的骄傲! 我为他点赞! 愿他这次中国之行, 收获满满! - 情感: 积极
评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极
评论: 全球最睿智的两位外长即将会谈 - 情感: 积极
评论: 欢迎老朋友来华。中俄联手, 世界有希望! - 情感: 消极

图 4-5 Word2vec 算法的支持向量机模型进行新闻评论情感分析

通过最终系统中爬取的凤凰新闻用户评论数据发现, 该模型正确率已经提高并且消极和积极识别能力已经增强。最终人工测试 157 条数据, 模型输出的预测已有 111 条分析正确。推算最终的模型的准确率大概在 70.7%左右。联系上下文已经有了成效。模型已经可以发现对一些长文本, 需要联系上下文的语句也做出很好的情感识别, 这也是本次不同特征工程 and 不同传统机器学习方法对新闻评论文本情感分析的最后一种方法。

4.5 新闻评论情感分类结果分析与总结

本节为本章节所有新闻评论机器学习算法结果的总结与分析。本章节主要采用了 5 种不同特征选择和分类算法的组合, 结果也不尽相同, 总体上也差强人意。主要使用的评价标准为精确率 (Precision), 召回率 (Recall) 和 F1 分数 (F1 Score)。首先新闻评论情感分类精确率最低的算法为“TF-IDF 算法的决策树新闻评论情感分类模型”, 算法精确率最高的为“Word2vec 算法的支持向量机新闻评论情感分析模型”, 各种模型在面对测试文本标注为“积极”的样例分类精确度和“消极”的样例分类精确率如下图 4-6 所示:

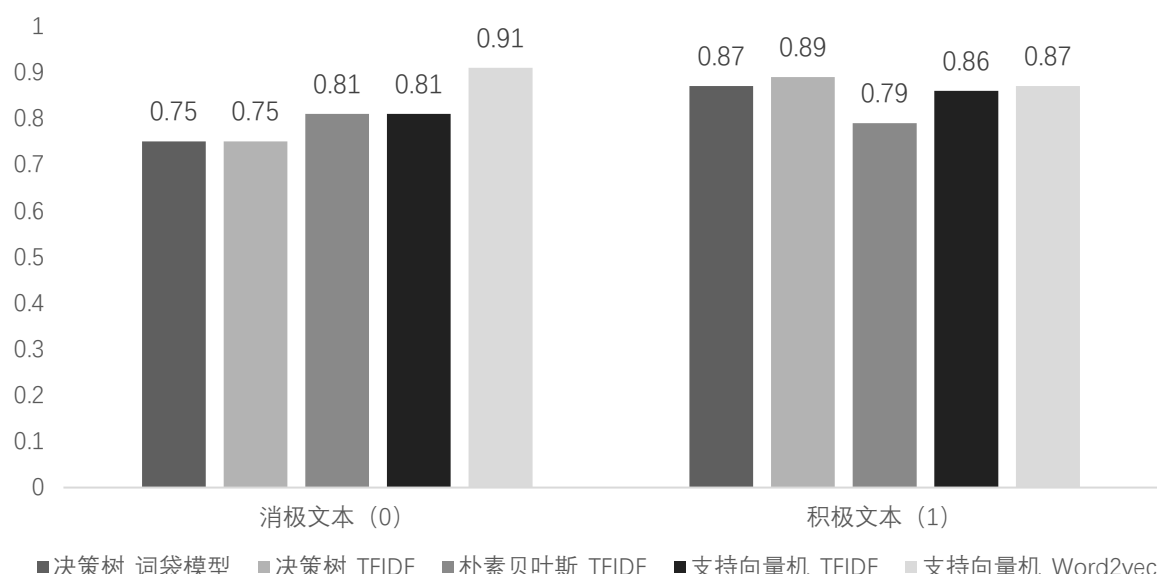


图 4-6 不同算法对测试文本预测精确率

由图可知，在消极测试文本中“支持向量机_Word2vec”精确率最高，达到了 0.91，而在积极测试文本中，“支持向量机_Word2vec”精确率也比较高，达到了 0.87，这也是因为支持向量机在高维空间样本中可以很好地进行分类。由于各种模型对训练和测试数据的拟合程度不同。在这里画出不同算法的准确率，召回率，和 f1 分数，分析不同模型预测正确率情况，如下图 4-7 所示：

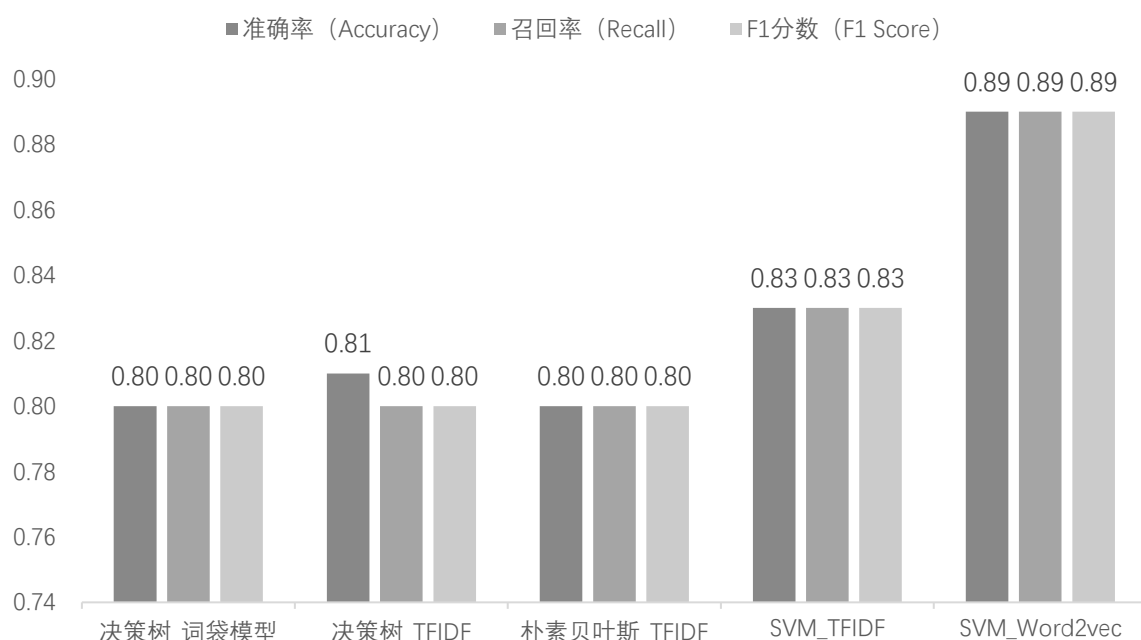


图 4-7 不同算法的准确率，召回率，和 f1 分数

对于“决策树_词袋模型”，“决策树_TFIDF”，“朴素贝叶斯_TFIDF”，“SVM_TFIDF”，“SVM_Word2vec”五种不同算法的准确率与人工验证准确率如下图 4-8 所示：

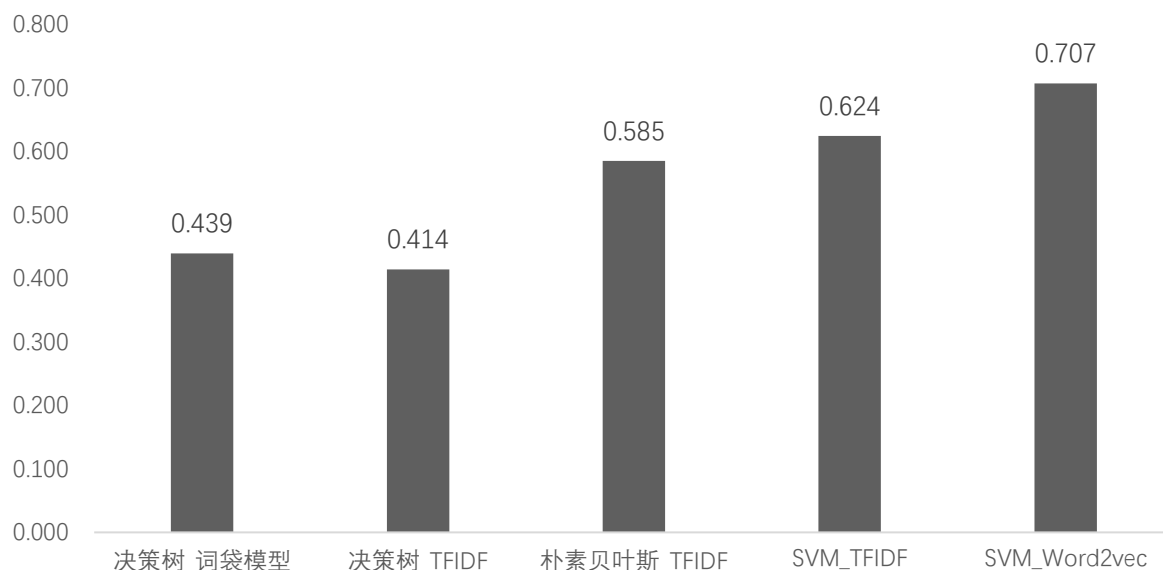


图 4-8 人工验证准确率

由此可见，新闻评论情感分析模型的准确率差异可以归因于多个核心因素，深刻反映了模型设计、特征工程以及数据特性的相互作用与影响。首先，针对决策树模型（尤其是结合词袋模型和 TF-IDF 特征表示时）准确率偏低的现象，其原因可以进一步细化如下：

（1）过拟合风险：决策树模型在处理高维度特征空间时，如文本数据的庞大词汇表，容易构建出过于复杂的决策边界，从而在训练数据上表现优异，却难以泛化至未见数据。

（2）不平衡数据敏感性：新闻评论数据集经常面临类别不平衡问题，决策树对此较为敏感，可能导致模型偏向于多数类，牺牲了对少数类的识别能力。

（3）模型表达能力限制：作为较简单的模型，决策树可能无法充分捕捉文本中复杂的语法结构和语义关系。

（4）特征表示局限性：词袋模型仅统计词汇频次，忽略了词汇间的顺序及上下文关联，而 TF-IDF 虽提升了关键词的重要性考量，但仍未能解决词序缺失的问题。

（5）特征稀疏性挑战：TF-IDF 特征往往导致高维稀疏向量，这可能降低了决策树模型的效率和性能。

相比之下，朴素贝叶斯模型（使用 TF-IDF 特征）之所以能展现较好的性能提升，原因在于：

（1）高效处理文本数据：朴素贝叶斯基于特征独立假设，尽管这一假设在真实语言中不尽准确，但在大量文本数据分类中仍能有效工作。

（2）TF-IDF 的优化效果：TF-IDF 增强了特征的区分度，使得朴素贝叶斯模型能够更好地利用关键信息，提升分类精度。

（3）计算效率与可扩展性：朴素贝叶斯算法的计算简便，尤其适合处理大规模文本

数据集，确保了在实践中的高效应用。

至于 SVM 模型（尤其是 Word2Vec 特征）展现出的高准确率，则得益于：

（1）高维空间处理能力：SVM 擅长处理高维特征空间，能在复杂的数据结构中寻找最优决策边界，这对于文本数据的分类至关重要。

（2）核函数的灵活性：通过核技巧，SVM 能够应对非线性可分数据，这对于理解文本中蕴含的复杂模式和关系尤为关键。在本文中就使用 TF-IDF 算法搭配非线性 SVM 核函数（RBF），Word2vec 模型搭配 SVM 线性核函数进行分类，均取得了不错的结果。

（3）Word2Vec 的语义优势：Word2Vec 提供的词嵌入不仅能捕捉词汇的共现关系，还能体现语义相似性，增强模型对文本深层意义的理解。

（4）SVM 与 Word2Vec 的协同效应：两者的结合不仅提高了模型对文本语境的把握能力，还促进了对复杂情感细微差别的精准捕捉，从而极大提升了情感分析的准确度。

而且几乎所有模型都出现了过拟合的情况，分析主要原因可能在于训练数据与测试数据差别过大，微博评论数据并不能很好的代表新闻评论数据，而没有捕捉到真实新闻评论数据的真实分布。

综上，不同模型和特征表示方法的选择，以及它们如何与特定数据集特征相适应，共同决定了新闻评论情感分析模型的性能表现，强调了在模型训练阶段深入理解数据特征和参数适配的必要性。最终该新闻评论情感分析模型准确率不高，只有 70% 左右。原因有很多，主要原因是本人能力有限，算法设计不完善，还有很大的上升空间。目前优秀的传统机器学习情感分类模型的准确率能达到 80% 甚至 90%，但由于自身能力有限，无法读懂相关论文，没能深入了解每个特征和参数的作用，理解算法思想比较吃力，没能实现更加优秀的新闻评论情感分析模型。

第5章 新闻评论情感分析系统实现

新闻评论原始数据集经过数据预处理,特征抽取,模型训练等操作之后,得到了最终准确率较高的新闻评论情感分析模型。系统实现方面可以根据用户自定义输入的凤凰新闻网 URL,自动调用爬虫模块爬取该新闻下所有用户评论数据,然后用户可以自主选择已经保存好的新闻评论情感分析模型对爬取数据进行情感分析,并把用户评论数据情感分析结果,用户词云图,每日评论数量变化折线图输出到前端页面上,可以让用户更直观更清晰地了解当前新闻的大众评论情感倾向。

5.1 爬虫模块

首先,通过 input 函数提示用户输入 URL。观察凤凰新闻页面 URL 发现,不同新闻网页 URL 最后 11 位为凤凰新闻不同新闻文章的唯一识别码,所以使用 get_lasteleven 函数提取 URL 的最后 11 个字符。设置 HTTP 请求头部信息,创建获取评论信息的接口,在凤凰新闻页面中“发表评论”按钮,进入“全部评论页面”,点击 F12 检查网页按钮,在网络下,可以找到评论信息 GET 请求的唯一接口,如图 5-1 所示:

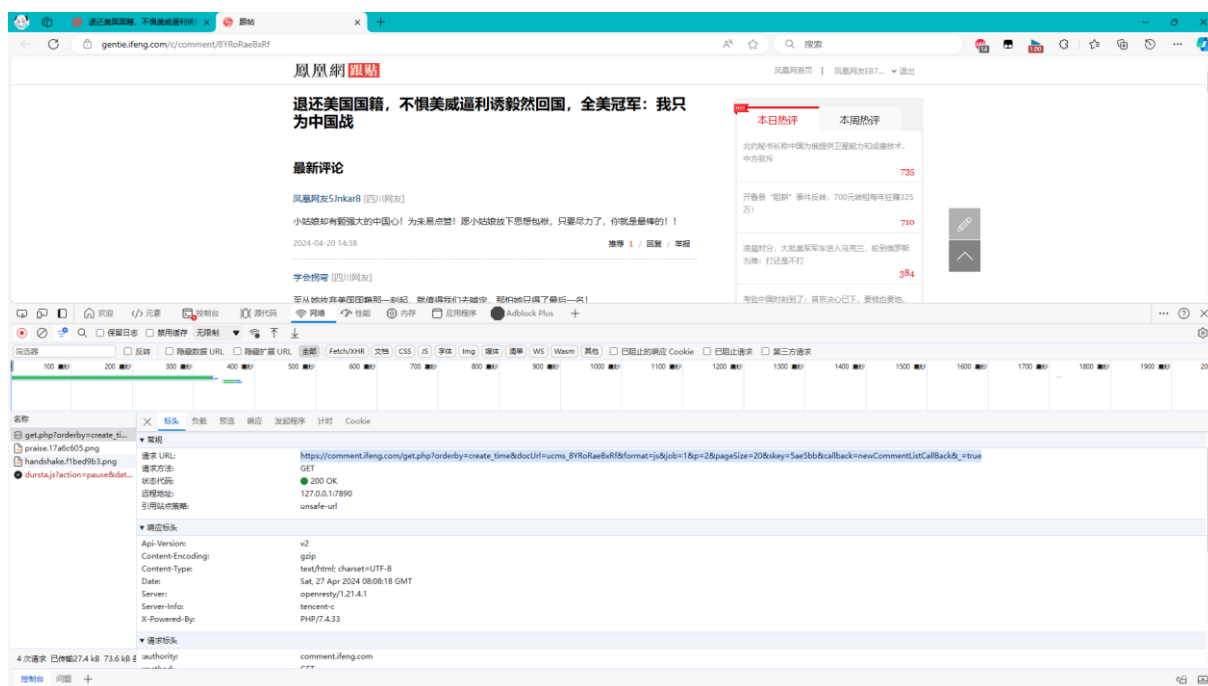


图 5-1 评论信息 GET 请求的唯一接口

创建一个空的 DataFrame 来存储所有评论。使用 while 循环和 create_page 函数来遍历每一页评论。在循环中,使用 fetch_comments 函数获取当前页的评论。使用 fetch_comments 函数返回解析后的 JSON 中的评论数据,需要转换为 utf-8 编码。如果某一页返回的评论为空,表示已经到达最后一页,循环结束。将每页获取的评论转换为 DataFrame,然后追加到 all_comments 中。循环结束后,使用 get_unique_filename 函数获取唯一的文件名,并将所有评论保存到 CSV 文件中。

5.2 词云模块

词云图（Word Cloud 或 Tag Cloud）是一种视觉化工具，它通过改变字体大小、颜色或排列方式来展示文本数据中各词汇的频率或重要性，从而直观地反映出文本的主要内容和焦点。通过调整词云中词汇的大小来反映情感强度，较大的词汇通常代表在评论中频繁出现且情感倾向显著的词汇。这样可以直观地展示出评论者对于新闻主题的主要情感倾向，比如正面评价或负面批评。结合情感分析结果，词云图可以帮助分析师深入探索评论中特定情感的来源，例如，将正面和负面评论分开制作词云，分别展示积极和消极的情感焦点，从而对评论内容有更细致的理解。

使用 WordCloud 类创建词云实例，设置字体路径、尺寸和背景颜色。调用 generate(text) 方法生成词云，并通过 to_image() 转换为图像。将词云图像转换为 Tkinter 兼容的格式，并通过 label 标签在 GUI 中显示。词云生成示例如下图 5-2 所示：



图 5-2 词云生成示例

5.3 折线图模块

折线图作为一种数据可视化工具，在新闻评论分析领域扮演着至关重要的角色。折线图能够清晰展示新闻评论数量随时间推移的变化趋势，揭示了公众对不同新闻事件的即时反应和兴趣持续周期。通过观察折线图上的波动情况，分析人员可以了解公众对某个新闻事件的关注是否持续，以及他们的参与度是否随着时间的推移而减弱。这对于评估新闻事件的影响力和持久性非常重要。折线图作为新闻评论分析的重要工具，能够帮助分析人员清晰地展示新闻评论数量随时间推移的变化趋势，从而深入理解公众对特定新闻事件的关注度如何变化。通过观察折线图上的高峰和低谷，分析人员可以快速定位到引起公众广泛关注或激烈讨论的具体新闻事件或时间点，这对于分析公众的反应和兴趣持续周期具有重要意义。

使用 FigureCanvasTkAgg 将 Matplotlib 图表嵌入到 Tkinter GUI 中，并显示。每日评论数量变化折线图生成示例如下图 5-3 所示：

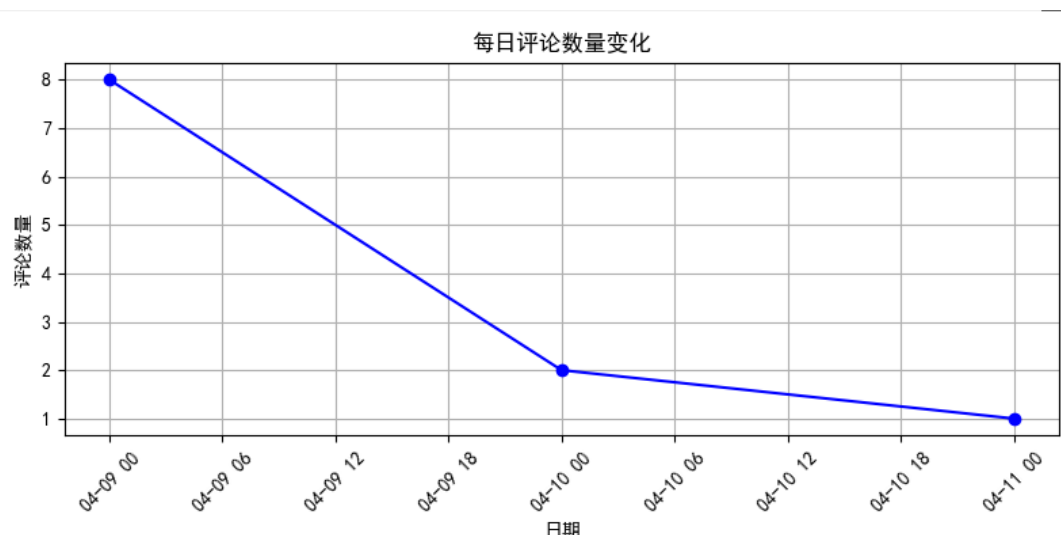


图 5-3 每日评论数量变化折线图生成示例

5.4 前端页面

本文系统前端页面使用 TK inter 实现，Tkinter 作为 Python 的内置图形用户界面 (GUI) 工具包，为开发者提供了一种快速、简便且高效的方式来构建跨平台的桌面应用程序。其普及度高，作为标准库的一部分几乎在所有 Python 环境中均可用，这使得它无需安装额外框架即可运行，极大地简化了部署和维护工作。Tkinter 以其易学性和丰富的文档及社区支持而著称，使得新手可以迅速上手。而且，Tkinter 支持多操作系统运行，确保了在不同平台上的应用能够提供一致的用户体验。它提供了灵活的布局管理和丰富的控件库，方便开发者根据需求定制个性化界面和交互逻辑。尽管在界面美观度和处理极端性能优化方面可能不如一些现代框架，但 Tkinter 以其事件驱动的编程模型和可视化开发工具的支持，仍然是一个适合快速开发小型到中型应用的强大工具。

用户在界面的输入框输入凤凰新闻链接或者输入本地已有评论数据集，数据集中需确保评论数据列索引为“comment_contents”，时间列表索引为“create_time”。输入凤凰新闻在线新闻链接或文件地址之前前端页面如图 5-4 所示：



图 5-4 用户输入界面

输入在线凤凰新闻链接（退还美国国籍，不惧美威逼利诱毅然回国，全美冠军：我只为中国战 凤凰网 (ifeng.com)）之后，在下拉框中选择想要使用的已经保存好的新闻评论情感分析模型，默认为“决策树_词袋模型”，这里选择准确率最高的“SVM_Word2vec”进行接下来的演示，模型选择界面如图 5-5 所示。



图 5-5 选择新闻评论分析模型

确定网页URL或者本地数据集路径及选择的新闻评论分析模型无误之后，点击“分析情感”按钮，如下图 5-6 所示：

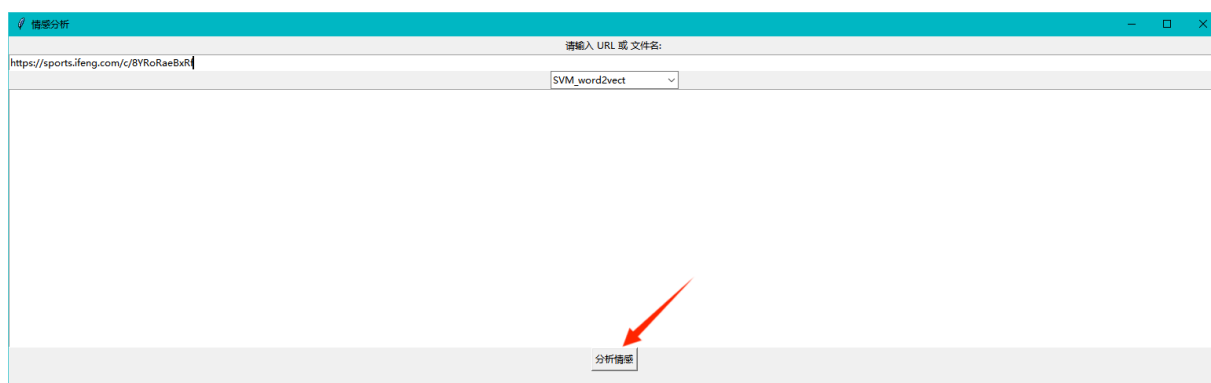


图 5-6 “分析情感”按钮

系统会自动根据新闻URL调用爬虫模块爬取该新闻所有用户评论，并保存在项目目录下，系统会生成一个名为“test_”加上时间日期的csv评论数据文件，数据集中有“uname”, “create_time”, “comment_content”三个字段，之后为保证特征数量一致，采用和训练模型时一样的“分词”，“去除停用词”，“正则清洗”手段对刚刚爬取的用户评论进行数据预处理，之后根据用户选择的文本分析模型和向量器自动对爬对处理好的文本进行向量化处理，并调用已经保存好的模型对文本进行预测。稍作等待即可在输出框中输出该新闻下所有用户评论数据和模型情感预测。如图 5-7，图 5-8 分别展示了运用模型准确率最低的“决策树_TFIDF”新闻评论情感分析模型和准确率最高的“SVM_Word2vec”新闻评论情感分析模型选用同一篇凤凰网在线新闻（退还美国国籍，不惧美威逼利诱毅然回国，全美冠军：我只为中国战 凤凰网 (ifeng.com)）进行在线新闻评论情感分析，并输出用户每日评论数据折线图和用户评论词云图，分析后的系统页面效果如下图 5-7，图 5-8 所示：



图 5-7 “决策树 TFIDF” 新闻评论情感分析模型



图 5-8 “SVM Word2vec” 新闻评论情感分析模型

页面包括四个部分，分别是最上方的链接输入框，中间的模型选择下拉栏，中间的新闻评论情感分析预测输出框，下方的每日评论数量变化折线图和词云图。

对比图 5-7 和图 5-8，最大的区别就在于中间的新闻评论情感分析预测输出框，由于选择的新闻评论情感分析模型不同，输出的预测结果也不尽相同，下方的每日评论数量变化折线图和词云图不会有很大区别。

结束语

本论文旨在研究基于机器学习的新闻评论情感分析方法。通过使用 Python 语言进行文本挖掘，对凤凰新闻用户评论数据进行中文情感分析，论文选择了三种特征工程方法：词袋模型、TF-IDF 算法和 Word2vec，以及三种不同的机器学习分类算法：决策树、朴素贝叶斯和支持向量机，以对新闻评论进行情感倾向性分类。此外，本项目还利用爬虫技术自动获取用户评论数据，并通过已训练好的情感分析模型对这些数据进行自动分析，生成“每日评论数量变化”的折线图和反映情感倾向的词云图，实现了自动获取、分析和可视化用户评论数据的功能。

不足之处在于，尽管采用了多种特征工程和机器学习算法，但模型在面对语义复杂或依赖上下文理解的评论时，表现出了局限性，准确率未能达到较高水平。

未来研究可以考虑结合深度学习技术进一步提升情感分析的准确度和效率。同时，可以探索更多自然语言处理的技术来改善模型性能，如引入更先进的词嵌入技术或使用预训练模型。此外，对于系统实现部分，可以考虑优化前端页面的用户交互体验，增强系统的可用性和用户体验。还可以考虑将模型部署到云平台，以支持大规模数据分析和实时情感倾向监控。

致 谢

岁月如梭，白驹过隙，当此论文完成之际，吾心潮澎湃，思绪万千。回首往昔，求学之路漫漫，幸得诸师友之帮助，方能克难前行，终至今日之成。是以，谨以此简札，聊表寸心之谢忱，虽言不尽意，然情深意长。

首先，饮水思源，感念吾师。X 老师，德高望重，学识渊博，犹如松柏之坚韧，兰芷之幽香，不仅授业解惑，更指点迷津，示吾立身治学之道。先生之教诲，如春风化雨，润物无声，使吾学术之籽得以破土成芽。师恩浩荡，终生难忘。

其次，家庭港湾，感恩亲慈。双亲犹如巍巍泰山，默默支撑，无论寒暑易节，始终给予最坚实的后盾与最温暖的关怀。家之温馨，是我勇往直前的不竭动力，此恩此爱，山高水长，无以为报，唯有更加勤勉，以期不负所望。

再者，同窗数载，感谢挚友。诸君或激扬文字，或慎思明辨，相互砥砺，共赴书山之路。犹忆灯火阑珊夜读，笑语中智识交锋，友谊之花于书香氤氲里静默盛开。此情此景，历历在目，诚所谓“海内存知己，天涯若比邻”。

此外，亦须特别鸣谢那些在我求学途中辛勤耕耘的普通任课教师。他们虽未如导师般深入指导，却每授课之际皆为知识之基石。耐心讲授专业知识，严谨阐释学术理论，承蒙指引，由白丁而至学术之门。

最后，向所有在本文完成过程中提供帮助与启发的文献作者、机构教程表达由衷的敬意与感谢。纸短情长，难以尽述心中感激。吾将铭记此段历程，怀揣感恩之心，踏入新的学途，期许未来能以所学回馈社会，不忘初心，光大知识之火。

时维甲辰仲夏，于东华理工大学堂，谨记，学疏才浅之学子 XXX，敬上。

参考文献

- [1] 卜晓阳,蔡岩,王宗伟,赵郭焱.基于 C5.0 决策树算法的电力营销数据挖掘[J].微型电脑应用,2022,38(01):23-26.
- [2] 彭宇翔,文继芬,李皓,刘涛,唐辟如,郭茜.基于决策树模型的贵州降雹识别研究[J].中低纬山地气象,2021,45(06):99-101.
- [3] 贾晓帆,何利力.融合朴素贝叶斯与决策树的用户评论分类算法[J].软件导刊,2021,20(07):1-5.
- [4] 邓晓林,陈毅红,王登辉.大数据环境下决策树的研究[J].太原师范学院学报(自然科学版),2021,20(02):47-57.
- [5] 冷婷,叶仁玉,李沅静.基于因子分析朴素贝叶斯方法的新闻文本分类[J].安庆师范大学学报(自然科学版),2024,30(01):47-51.
- [6] 张财,马自强,闫博.基于机器学习的政务微博情感分析模型设计[J].计算机工程:1-14.
- [7] 陈冬林,吴天昊,吴江,徐书情.基于 word2vec 的内容过滤科技成果推荐模型研究[J].武汉理工大学学报(信息与管理工程版),2023,45(04):599-606.
- [8] Pushpam Kumar Sinha. Modifying one of the Machine Learning Algorithms KNN to Make it Independent of the Parameter k by Re-defining Neighbor[J]. International Journal of Mathematical Sciences and Computing(IJMSC), 2020, 6(4) : 12-25.
- [9] Jayasree Saha and Jayanta Mukherjee. CNAK : Cluster number assisted K-means[J]. Pattern Recognition, 2021, 110.
- [10] Li Teng and Dou Yong. Representation learning on textual network with personalized PageRank[J]. Science China Information Sciences, 2021, 64(11)
- [11] Cross-subject emotion recognition using hierarchical feature optimization and SVM with multi-kernel collaboration[J].Physiological measurement. Volume 44 , Issue 12 . 2023
- [12] Heart Disease Prediction System using Ensemble of Machine Learning Algorithms [J].Recent Patents on Engineering. Volume 15 , Issue 2 . 2021. PP 130-139
- [13] Yuval E ,Maytal T C .From density functional theory to machine learning predictive models for electrical properties of spinel oxides[J].Scientific Reports,2024,14(1):12150-12150.